DECLARATION

I, Yoshito Yamada, c/o YAMADA PATENT OFFICE of The Tanabe Bldg., 6-6, Fushimimachi 2-chome, Chuo-ku, Osaka-shi, Osaka, Japan, declare that I am the translator of the documents attached, which are to the best of my knowledge and belief a true and correct translation of Japanese Patent Application No. **11-282592**.

DATE: January 25, 2008

Signature of translator

Yoshito Yamada

[Document name] SPECIFICATION

[Title of the invention] Portable game apparatus having an acceleration sensor and information storage medium storing a game program

[Claims]

[Claim 1]

A portable game apparatus having an acceleration sensor, comprising:

a display for displaying a game scene;

a housing to be held by the hand of a player and having the display arranged on one main surface thereof;

an acceleration sensor provided related to the housing;

neutral-position set means for setting a reference inclination of the housing;

temporary storing means for temporarily storing as neutral position data an output value of the acceleration sensor corresponding to a neutral position;

correcting means for correcting the output value of the acceleration sensor with the neutral-position data; and

game control means for changing a display state of the game scene to be displayed on the display based on an output value of the correcting means.

[Claim 2]

A portable game apparatus having an acceleration sensor according to claim 1, wherein the neutral-position set means includes an operation key to be operated by the player and a program to write, as the neutral position data, an output value of the acceleration sensor corresponding to a tilt of the housing in operating the operation key by the player to the temporarily storing means.

[Claim 3]

A portable game apparatus having an acceleration sensor according to claim 1 or

2, wherein the neutral position setting means includes

sight display means to display, on the display, a sight moving in accordance with a tilt of the housing, and

target coordinate display means to display on the display a target coordinate that the sight is to position when the housing is in a proper inclination,

an output value of the acceleration sensor corresponding to a tilt of the housing when the sight overlaps with a target coordinate being set as the neutral-position data.

[Claim 4]

An information storage medium to be used for a portable game apparatus including a display to display a game scene and a housing to be held by the hand of a player and having the display arranged on one main surface thereof, an information storage medium storing a game program comprising:

an acceleration sensor output read program for reading an output value of the acceleration sensor;

a neutral-position set program for setting a reference inclination of the portable game apparatus;

a write program for writing as neutral position data an output value of the acceleration sensor corresponding to the neutral position to the temporary storage means;

a correction program for correcting an output value of the acceleration sensor by the neutral-position data; and

a game control program for changing a display state of the game scene to be displayed on the display based on an output value of the correction program.

[Claim 5]

An information storage medium storing a game program according to claim 4, wherein the acceleration sensor is provided related to the housing.

[Claim 6]

An information storage medium storing game program according to claim 4, wherein said information storage medium is a cartridge to be removably attached in the housing of the portable game apparatus, and the acceleration sensor being accommodated in the cartridge.

[Claim 7]

An information storage medium storing a game program according to claim 4, wherein the portable game apparatus includes an operation key, and

the write program writing, as neutral position data, an output value of the acceleration sensor corresponding to a tilt of the housing in operating the operation key by the player to the temporary storing means.

[Claim 8]

An information storage medium storing a game program according to claim 4 or 7, wherein the neutral-position set program includes

a sight display program to display on the display a sight moving in accordance with a tilt of the housing, and

a target coordinate display program to display on the display a target coordinate that the sight is to position when the housing is in a proper tilt,

a tilt of the housing when the sight overlaps with a target coordinate being set as the neutral-position data.

[Detailed description of the invention]

[0001]

[Industrial field of utilization]

This invention relates to a portable game apparatus having an acceleration sensor and information storage medium storing a game program and, more particularly, to a

portable game apparatus for detecting a tilt or movement of the portable game apparatus by an acceleration sensor to utilize as operation information in game control and to an information storage medium storing a game program.

[0002]

[Prior art]

Conventionally, the portable game apparatus has provided with an input device, such as a cross key and button so that the player could operate game characters and select commands and the like by operating the input device while holding the portable game apparatus by the both hands.

The prior arts utilizing a tilt sensor or acceleration sensor as a game input device are disclosed in Japanese Patent Laid-open No. 181630/1993, Japanese Patent Laid-open No. 198075/1994, Japanese Patent Laid-open No. 191953/1996 and Japanese Patent Laid-open No. 21000/1998. In these prior arts, a tilt sensor or acceleration sensor is added to the controller provided independent of the game apparatus main body to be used connected to a video game apparatus in order to detect a movement or tilt of the controller. The player is allowed to control the game characters displayed on the television screen by moving or tilting the controller.

[0003]

[Problem to be solved by the invention]

In the foregoing prior arts, the acceleration sensor or the like is added to the controller of the video game apparatus that the display of a television receiver or the like is separately provided from the game apparatus main body, it is impossible to apply as it is to a portable game apparatus. This is because, although the portable game apparatus includes a display such as an LCD so that a game can be played by holding the housing at opposite ends by both hands, it is usual for the player to hold the portable game apparatus

with inclination relative to a horizontal direction so that LCD display can be readily seen during game play. Where the portable game apparatus is tilted relative to the horizontal direction, the acceleration sensor will detect an acceleration even if there is no acceleration input (or tilt input) by the intention of the player. Also, the tilt of the portable game apparatus held by the player refers according to various factors, such as player individual difference, light source position such as of a light, game content the position of a player in game play, etc.

[0004]

Therefore, it is a primary object of this invention to provide, in a portable game apparatus having an acceleration sensor, a portable game apparatus capable of setting a tilt of the portable game apparatus held by a player as a neutral position so that the player can play a game while holding it with a tilt easy to play the game.

It is another object of this invention to provide, in an information storage medium used in a portable game apparatus having an acceleration sensor, an information storage medium storing a game program capable of setting a tilt of the portable game apparatus held by a player in game play as a neutral position so that the player can play a game while holding with a tilt easy to play.

It is further object of this invention to provide a game cartridge simple in structure and cheap in price but capable of processing based on an output of an acceleration sensor even where the portable game apparatus is already under marketing.

[0005]

[Means for solving the problem]

A portable game apparatus having an acceleration sensor of a first invention (invention described in claim 1) comprises a display for displaying a game scene, a housing to be held by the hand of a player and having a display arranged in one main

surface thereof, an acceleration sensor provided related to the housing, neutral-position set means for setting a reference tilt of the housing, temporary storing means for temporarily storing an output value of the acceleration sensor corresponding to a neutral position as neutral position data, correction means for correcting an output value of the acceleration sensor by neutral-position data, and game control means for changing a display state of a game scene to be displayed on the display based on an output value of the correction means.

[0006]

Herein, the game scene is a scene constituted by a game character (player character, ally character, enemy character and stationary character, etc.), an environment that the game character exists ("water" or the like where the game character exists under water) and a background, and the game control means changes a display state of them. For example, the player character or enemy character is displayed changing in movement or shape, and the environment or background is changed in display or scrolled.

[0007]

An information storage medium storing a game program of a second invention (invention described in claim 4) is an information storage medium to be used for a portable game apparatus including a display to display a game scene and a housing to be held by the hand of a player and having the display arranged in one main surface thereof, and comprises an acceleration sensor output read program to read an output value of the acceleration sensor, a neutral position set program to set a reference tilt of the portable game apparatus, a write program to write, as neutral position data, an output value of the acceleration sensor corresponding to the neutral position to a temporary storing means, a correction program to correct an output value of the acceleration sensor by the neutral position data, and a game control program to change a display state of a game scene to be

displayed on the display based on an output value of the correction program.

[0008]

[Operation]

In a first invention of a portable game apparatus having an acceleration sensor, before game play, a reference inclination of the portable game apparatus is set by the neutral position set means. The reference inclination may be set to an arbitrary inclination by the player or set to a predetermined inclination by a program. Otherwise, any one of a plurality of inclinations previously determined by program may be selected by the player. For example, where the player sets it to an arbitrary inclination, the player holds the portable game apparatus with an inclination easy to play a game and operates the operation key. Thereupon, an output value of the acceleration sensor upon operating the operation key is written and held as neutral position data in the temporary storage means. In the game processing, the output value of the acceleration sensor is corrected based on the neutral position data by the correction means, and thereafter the game control means changes the display state of game scene. Incidentally, the neutral position set means may be provided to set a neutral position in the course of game play.

[0009]

In a second invention of an information storage medium storing a game program, before game play, the neutral-position set program sets a reference inclination for the portable game apparatus. The acceleration-sensor-output read program reads out an acceleration-sensor output value corresponding to the neutral position, and the write program writes it as neutral position data to the temporary storage means. This temporary storage means may be storing means in the information storage medium or storing means in the portable game apparatus. In game play, although the acceleration-sensor-output read program reads out an acceleration-sensor output value to perform game processing,

the acceleration-sensor output value is corrected based on the neutral position data by a correction program and then the game control program changes a display state of game scene.

[0010]

[Effect of the invention]

According to this invention, in the portable game apparatus having an acceleration sensor, the player is allowed to set an inclination of the portable game apparatus held by the player as neutral position so that the player can hold it with an inclination easy to play thereby enabling game play.

Furthermore, according to this invention, even where the portable game apparatus is already under marketing, the processing based on an output of the acceleration sensor is made possible with a simple structure at low cost.

[0011]

[Detailed description of the invention]

Embodiments of the present invention will be explained below with reference to the drawings.

Figure 1 is an outside view of a portable game apparatus. The portable game apparatus includes a game machine main body 10 and a game cartridge (hereinafter referred merely to as "cartridge") 30 to be unloadably loaded on the game machine main body 10. The cartridge 30, when loaded on the game machine main body 10, is put in electrical connection to the game machine main body. The game machine main body 10 is provided with a housing 11. The housing 11 includes therein a board having circuits configured as shown in Figure 3, hereinafter described. The housing 11 has, on one main surface, a LCD 12 and operation keys 13a – 13e and, on the other surface, a hole (cartridge insertion hole) 14 formed to receive a cartridge 30. A connector 15 is provided

on a side surface, to allow connection with a communication cable for communication, as required, with other portable game apparatuses.

[0012]

Figure 2 is an illustrative view showing a relationship between the portable game apparatus and XYZ axes thereon. In a state the portable game apparatus is positioned with the LCD 12 directed up and the operation keys positioned toward this, an X-axis is taken in a horizontal direction of the portable game apparatus (a plus direction taken rightward), an Y-axis is in a vertical direction (a plus direction taken depthwise), and a Z-axis is in a thickness direction (a plus direction taken upward).

[0013]

Figure 3 is a block diagram of the portable game apparatus. The game machine main body 10 incorporates a board 27 therein. The board 27 is mounted with a CPU 21. The CPU 21 is connected with a LCD driver 22, an operation key 13, a sound generator circuit 23, a communication interface 24, a display RAM 156 and a work RAM 26. The sound generator circuit 23 is connected with a speaker 16. The communication interface 24 is to be connected to another portable game apparatus 40 through a connector 15 and communication cable 50. Note that, although the communication method with the other portable game apparatus 40 was shown by a method using the communication cable 50, it may use radio communication, handy phone or the like.

[0014]

The cartridge 30 incorporates a board 36. The board 36 is mounted with a program ROM 34 storing a game program and game data, hereinafter described with reference to Figure 21, and a backup RAM 35 storing a game data, hereinafter described with reference to Figure 24. In addition to these storage means, the cartridge 30 includes, as one example of acceleration detecting means, an XY-axis acceleration sensor 31 to

detect accelerations in X-axis and Y-axis directions and a Z-axis contact switch 32 to detect an acceleration in a Z-axis direction. Also, the cartridge 30 includes a sensor interface 33 as an interface to the acceleration detecting means. Where using a triaxial acceleration sensor capable of detecting accelerations in all the X-axis, Y-axis and Z-axis directions, the Z-axis contact switch 32 will be unnecessary. Incidentally, the biaxial acceleration sensor (XY-axis acceleration sensor) is more inexpensive. Because this embodiment does not require high accuracy of acceleration detection in the Z-axis direction, explanation will be on the case using a Z-axis contact switch 32 simple in structure and cheap in price.

[0015]

The program ROM 34 is stored with a game program to be executed by a CPU 21. The work RAM 26 is stored with temporary data required to execute the game program. The backup RAM 35 is to store game data to be kept memorized even where a power to the portable game apparatus be turned off. The display data obtained through executing the game program by the CPU 21 is stored in the display RAM 25, which can be displayed on the LCD 12 through a LCD driver 22. Similarly, the sound data obtained through executing the game program by the CPU 21 is delivered to the sound generator circuit 23 so that sound is generated as effected sound through the speaker 16. The operator can input by operating the operation key 13. However, the operation key 13 is auxiliary one as far as the present embodiment is concerned. The player is allowed to input-operate by tilting or moving the portable game apparatus. The tilt, movement and impact to the portable game apparatus during game operation are to be detected by the XY-axis acceleration sensor 31 and Z-axis contact switch 12. The CPU 21 can execute the game program by utilizing the output values of the acceleration detecting means,.

[0016]

For a game with using a plurality of portable game apparatuses, the data obtained through executing a game program by the CPU 21 is delivered to the communication interface 24 and then sent to another portable game apparatus 40 via the connector 15 and communication cable 50. Meanwhile, the game data of the other portable game apparatus 40 comes to the CPU 21 through the communication cable 50, connector 15 and communication interface 24.

[0017]

Figure 4 is a detailed block diagram of the sensor interface 33. The sensor interface 33 includes an X counter 331, a Y counter 332, a count stop circuit 33, latches 334 and 335, a decoder 336 and a general-purpose I/O port 337. The X counter 331 counts pulses of a clock signal Φ based on an XY-axis output of the acceleration sensor 31. The Y counter 332 counts pulses of the clock signal Φ based on a Y-axis output. The count stop circuit 333 sends a count stop signal to the X counter 331 in response to a fall in an X-axis output of the XY-axis acceleration sensor 31, and a count stop signal to the Y counter 332 in response to a fall in the Y-axis output. The latches 334 and 335 hold respective values of the X counter 331 and the Y counter 332. The decoder 336 transmits a start/reset signal to the X counter 331, Y counter 332, latches 334 and 335. The general-purpose I/O port 337 is used to connect with an extension unit. The latches 334 and 335 also hold an output value of the Z-axis contact switch 32 ("0" or "1"). Specifically, a highest order bit of the latch 334, 335 is assigned to an output value of the Z-axis contact switch 32, while the remaining lower order bits are assigned to the values of the X counter and Y counter. The extension units connectable to the general-purpose I/O port 337 include a vibration unit which vibrates in relation to a game program providing a game with a realism feeling.

[0018]

Figure 5 is an illustrative view showing a principle that the sensor interface 33 measures a count value having a corresponding magnitude to an acceleration from an output of the acceleration sensor 31. The acceleration sensor 31 in this embodiment outputs a signal representative of an acceleration magnitude with a duty ratio changed with respect to one period of a waveform (period 1). It is shown in this case that the greater the ratio of a high level period (period 2 or period 3) within one period the greater an acceleration has been detected. Also, the acceleration sensor 31 outputs a magnitude of X-axis acceleration through its X-axis output and a magnitude of Y-axis acceleration through the Y-axis output.

[0019]

When a count start signal outputted from the decoder 336 becomes a low level, the X counter 331 detects a rise from low to high level in the X-axis output and then starts counting. Specifically, the X counter 331 inches up its count value each time a clock signal Φ is given, and stops the counting in response to a count stop signal sent from the count stop circuit 333. In this manner, the X counter331 counts on the clock signal Φ during a period (period 2) of between a rise of an X-axis output to a high level and a fall of same to a low level, immediately after the count start signal has become a low level. The Y counter 332, similarly, counts on the clock signal Φ during a period (period 3) of between a rise of the Y-axis output to a high level and a fall of same to a low level, immediately after the count start signal has become low level. In this manner, the X counter 331 holds a count value dependent upon a magnitude of an X-axial acceleration while the Y counter 332 holds a count value dependent upon a magnitude of a Y-axial acceleration. The values of the X counter 331 and Y counter 332 are held in the latch 334 and latch 335 so that the data of latches 334 and 335 can be read out by the CPU 21 through the data bus and utilized for a game program.

[0020]

The X counter 331 and the Y counter 332 each perform counting, for example, from "0" up to "31", wherein setting is made such that, with respect to a count value "15" as a reference (acceleration 0), - 2G (twice a gravity acceleration in a minus direction) is assigned by a count value "0" and 2G (twice the gravity acceleration in a plus direction) is by a count value "31". The CPU 21 reads in such a count value based on a game program wherein the count value "15" is read as "0", the count value "0" as "-15" and the count value "31" as "16". Accordingly, when the acceleration sensor 31 detects an acceleration in the minus direction, the CPU has a minus (-) reading value. When an acceleration in the plus direction is detected, the CPU has a plus (+) reading value.

[0021]

Figure 6 shows a structural of the contact switch 32. The contact switch 32 is structured by a spherical contact 321, contacts 322 and 323, and a box member 324 which are formed of a conductor. Specifically, the spherical contact 321 is movably held almost at a center of a space defined within the member 324. For this reason, the box member 324 has, in its inner bottom, a depression 324a at which the spherical contact 321 can be rested at almost the center. The box member 324 has, at above, sheet-formed contacts 322 and 323 having respective one ends formed with semicircular cutouts 322a and 323a. The sheet contacts 322 and 323, at their other ends, are secured to a board 36 with the one ends opposed to each other. The box member 324 is fixedly held by the board 36 in a state of hung through the contact 322 and 323. With this structure, if the cartridge 30 is powerfully moved in the Z-axis direction (in a plus or minus direction), the spherical contact 321 shown in Figure 7 is moved in the Z-axis direction within the box member 324 and contacts with the contacts 322 and 323 simultaneously. Thus, the contact 322 and the contact 323 are conducted through the spherical contact 321, thereby detecting an

acceleration input in the Z-axis direction. Based on a time for which the contact 322 and the contact 323 are in conduction, it is possible to detect a magnitude of acceleration in the Z-axis direction. Note that, when the cartridge 30 is moderately tilted, the spherical contact 321 moves in the box member 324 but does not short-circuit between the contacts 322 and 323, detecting no acceleration.

[0022]

Figure 8 shows an example of a game scene. In this game scene, there are displayed a ball 61 as one example of a player character, tortoises 62 as one example of an enemy character (non-player character; hereinafter abbreviated as "NPC"), and a wall 63 and hole 64 forming a maze. Because a game map is a virtual map that is broader than a display range on an LCD 12, LCD 12 can display only part of the game map so that scroll is made in accordance with the movement of the player character 61. Although three tortoises 62a – 62c are displayed as NPC on the LCD 12, there exist many of other tortoises in the game map. Also, there exist on the game map such lands as floors, ice surfaces, and under waters as areas where the ball 61 is movable.

The ball 61 is changed in its moving amount or direction by player's operation, such as tilting of or applying movement or impact to the portable game apparatus. The shape is changed as required. Specifically, as hereinafter described by referring to Figure 17 to Figure 20, movement control is made by tilting or input an impact or input an acceleration in the Z-axis direction to the portable game apparatus. Although the tortoise 62 is controlled of movement (moved by self-control) by the game program, it is moved or changed in shape where the player tilts, gives impact or input acceleration in the Z-axis direction to the portable game apparatus.

[0023]

Outlining this game, a player can manipulate the ball 61 on the game map with a

maze formed by the walls 63, and smashes the tortoises 62a - 62c as an example of NPC. A tortoise if smashed will be vanished or erased away. If all the tortoises are successfully vanished out of the game map, game clear is reached. There exist some holes 64 on the game map. If the ball 61 is fallen into the hole 64, one mistake is counted or the game becomes over.

[0024]

Figure 9 to Figure 15 are figures showing operation examples of the portable game apparatus utilizing the XY-axis acceleration sensor 31 and Z-axis contact switch 32. Figure 9 is a view showing a slide input in the X-axis direction. A movement (slide) in the X-axis direction is detected based on an X-axis output of the XY-axis acceleration sensor 31. Figure 10 is a view showing a tilt input about the X-axis. An inclination about the X-axis is detected based on a Y-axis of the XY-axis acceleration sensor 31. Figure 11 is a view showing a slide input in the Y-axis direction. A movement (slide) in the Y-axis direction is detected based on a Y-axis of the XY-axis acceleration sensor 31. Figure 12 is a view showing a tilt input about the Y-axis. A tilt about the Y-axis is detected based on an X-axis output of the XY-axis acceleration sensor 31. Figure 13 is a view showing an impact input in the X-axis direction. Although an acceleration input in the X-axis is outputted from an X-axis output of the XY-axis acceleration sensor 31, where this output value is a constant value or greater, it is assumed that there is an impact input. Figure 14 is a view showing an impact input in the Y-axis direction. Although the acceleration input in the Y-axis direction is outputted from a Y-axis output of the XY-axis acceleration sensor 31, where this output value is a constant value or greater, it is assumed that there is an impact input. Figure 15 is a view showing an acceleration input in the Z-axis direction. The acceleration input in the Z-axis direction is detected by the Z-axis contact switch 32.

[0025]

Figure 16 to Figure 20 are views showing utilization examples concerning the above operation input. Figure 16 is a view showing a utilization method of a slide input (also an example of a game scene in a game-map select process hereinafter described with reference to Figure 39). Where displaying on the LCD 12 a part region of a virtual map greater than a display range of the LCD 12, the display region is scrolled by giving a slide input. Specifically, when a slide input is given in an X-axis plus direction, display is on a display region moved in the X-axis plus direction from the current display region. Similar processing is made for a slide input in the Y-axis direction. By thus processing the slide input, it is possible to have a feeling as if part of a broad world was seen through the LCD. Incidentally, in this embodiment, the slide input is utilized only in a game-map select process. The slide input is not utilized in the game process displaying game characters, etc. (game map scroll).

[0026]

Figure 17 is a view showing a utilization method of a tilt input about the X-axis. When there is a tilt input about the X-axis, the game character (player character and NPC) on a game scene is moved parallel in the Y-axis direction. Figure 18 is a view showing a utilization method of a tilt input about the Y-axis. When there is a tilt input about the Y-axis, the game character (player character and NPC) on a game scene is moved parallel in the X-axis direction. By thus processing the tilt input, it is possible to have a feeling as if the game character slid (rolled) over a display surface in accordance with a tilt of the portable game apparatus. Incidentally, on the game map, there mixingly exists the lands forming factors to change a moving state when a ball 61 is moved, i.e. floor, ice surface, under water and the like. The moving amount depending upon a tilt input is varied by the place where a game character exists. For example, in the case of an ice surface, the moving amount is great because of easy to slide while the moving amount in the case of

under water is less. In this manner, the moving state of the ball 61 is changed.

[0027]

Figure 19 is a view showing a utilization method of an impact input in the X-axis or Y-axis direction. When there is an impact input in the X-axis or Y-axis direction, different process is made from the tilt-input process (game character movement). For example, waves are caused in an environment that the game character is present. In the case of an impact input given in the X-axis plus direction, a wave is caused in the X-axis plus direction. Where an impact input is given in the X-axis minus direction, a wave is caused in the X-axis minus direction. This is similar to an impact input in the Y-axis direction. Also, a wave may be caused in a direction of a resultant vector having an X-axis directional vector component of an X-axis directional acceleration input and a Y-axis directional vector component of a Y-axis directional acceleration input. By this wave, the game character is moved flowing. The game character being flowed by the wave may be given uncontrollable.

[0028]

Figure 20 is a view showing a utilization method of an acceleration input in the Z-axis direction. When there is an acceleration input in the Z-axis direction, the ball 61 as a player character is changed to make jumping. During jump, the ball 61 will not move even if there is a tilt input. Also, when there is an acceleration input in the Z-axis direction, the tortoise 62 as NPC turns back (a tortoise upside down returns to the normal position). Because the tortoise upside down is easy to slide, moving process is made such that the moving amount due to the tilt input is greater as compared to that in the normal position.

[0029]

Figure 21 is a memory map of the program ROM 34. The program ROM 34 stores

a game program and game data to be executed by the CPU 21. The program ROM 34 concretely includes an object character data memory area 34a, a map data memory area 34b, an acceleration-sensor output value conversion table memory area 34c, a recommended-position sight-target coordinate memory area 34d and a game program memory area 34e. The object character data memory area 34a stores graphic data of the object characters. Because the object character has some poses (e.g. tortoise "normal position" and "upside-down position", etc.), for each character a plurality of ones of graphic data are stored for a plurality of poses. The map data memory area 34b stores map data on a game map basis and game-map-select maps. The game-map select map is virtual map data to be displayed on the LCD 12 during a game map select process hereinafter described with reference to Figure 39.

[0030]

The acceleration-sensor output value conversion table memory area 34c stores conversion tables to convert output values of the XY-axis acceleration sensor 31 and Z-axis contact switch 32, for utilization in a game program. The conversion tables includes a game map select table, a player character moving table and an NPC moving table. Also, the player character moving table includes tables for in-air, on-floor, on-ice and under-water, which are to be selected depending upon a land coordinate where a player character is present. The NPC moving table includes tables for normal position and upside-down position. The tortoise as NPC assumes states of normal and backside-down positions, depending upon which a table is to be selected. The details of the tables will be hereinafter described with reference to Figure 25 to Figure 32.

[0031]

The recommended-position sight-target coordinate memory area 34d stores coordinate data for a sight target coordinate (71 in Figure 36) to be displayed on the LCD

21 in a recommended-position set process, hereinafter described with reference to Figure 36 to Figure 38.

[0032]

The game program memory area 34e stores varies game programs to be executed by the CPU 21. Specifically, stored are a main program hereinafter described with reference to Figure 33, a 0G set program hereinafter described with reference to Figure 34, a neutral-position set program hereinafter described with reference to Figure 35, a recommended-position set program hereinafter described with reference to Figure 38, a game map select program hereinafter described with reference to Figure 39, a sensor output read program hereinafter described with reference to Figure 40, an object moving program hereinafter described with reference to Figure 41 to Figure 47, a collision program hereinafter described with reference to Figure 48, a screen scroll program hereinafter described with reference to Figure 51, an NPC self-controlled moving program and other programs.

[0033]

Figure 22 is a memory map of the work RAM 26. The work RAM 26 is to store temporary data for executing a game program by the CPU 21. Specifically, included are a neutral position data memory area 26a, an acceleration sensor memory area 26b, an impact input flag memory area 26c, a recommended-position-set-screen sight coordinate memory area 26d, a map select screen camera coordinate memory area 26e, a game map number memory area 26f and a character data memory area 26g.

The neutral position data memory area 26a stores neutral position data (NPx, NPy, NPz) to be set in a neutral-position set process hereinafter described with reference to Figure 35. This is data concerning a reference tilt of the portable game apparatus for playing a game.

The acceleration-sensor output value memory area 26b stores output values (INx, INy, INz) of the XY-axis acceleration sensor 31 and Z-axis contact switch 32 which are detected by the acceleration sensor 31 and contact switch 32 and to be read out through the sensor interface 33 in a sensor output read process of Figure 33.

The impact input flag memory area 26c stores an impact input flag (FS) that assumes 1 when equal to or greater than a constant value is the magnitude of resultant vector of a vector component in the X-axis direction taken of an acceleration input in the X-axis direction and a vector component in the Y-axis direction taken of an acceleration input in the Y-axis direction. The determination of impact input is executed in a sensor output read process of Figure 31.

[0034]

The recommended-position-set-screen target coordinate memory area 26d stores coordinate (Sx, Sy) for a sight (72 in Figure 36) to be displayed on the LCD 12 in a recommended-position set process hereinafter described with reference to Figure 36 to Figure 38.

The map select screen camera coordinate memory area 26e stores coordinates (Cx, Cy) at upper left corner of an LCD 12 display area in a game map select map which is to be displayed in a game map select process hereinafter described with reference to Figure 39.

The game map number memory area 26f stores corresponding number data (MN) to a game map having been selected by a player during a game map select process hereinafter described with reference to Figure 39.

[0035]

The character data memory area 26g stores, for each of the player characters and NPCs, moving acceleration data (Ax, Ay, Az), moving-acceleration change amount data

(dAx, dAy, dAz), velocity data (Vx, Vy, Vz), coordinate data (X, Y, Z), last-time coordinate data (Px, Py, Pz), current position status (SP) and pose numbers (PN).

The coordinate (Px, Py, Pz) in the last time is for returning to the last-time coordinate a player character or NPC when collided with a wall or the like. The current-position status data (SP) is data concerning a land at a coordinate where the player character is present. Based on this data, an acceleration-sensor output value conversion table (in-air, on-floor, on-ice, on-water) is to be selected. The pose number (PN) is data concerning a character state (pose) (e.g. tortoise normal and upside-down positions, etc.).

[0036]

Figure 23 is a memory map of the display RAM 25. The display RAM 25 is to temporarily store display data obtained through executing a game program by the CPU 21. The display RAM 25 has an object data memory area 25a, a scroll counter data memory area 25b and a map data memory area 25c. The object data memory area 25a stores data of the existing characters in the LCD 12 display area among all the characters to appear in a game. Specifically, stored area X-coordinates, Y-coordinates, character IDs, and pose numbers.

The scroll counter data memory area 25b stores a relative coordinate of an upper left corner of the LCD 12 display area of the game. The map data memory area 25c stores game map data of the game map in an area to be displayed on the LCD 12.

[0037]

Figure 24 is a memory map of the backup RAM 35. The backup RAM 35 stores 0G position data (ZGx, ZGy) to be set in a 0G set process hereinafter described with reference to Figure 34. The 0G position data is to cope with not to have a sensor output value of 0 because of the error possessed by the XY-axis acceleration sensor even when the portable game apparatus is held horizontal. A sensor output value when the portable

game apparatus is held horizontal is stored as 0G position data in the backup RAM 35, which in a game process is subtracted from a sensor output value.

[0038]

Figure 25 to Figure 32 illustrate in detail conversion tables stored in the acceleration-sensor output value conversion table memory area 34c of the program ROM 34. The tables store data, concerning utilization ways and correction of limiting a maximum values, etc., for utilizing, in game processing, sensor output values (INx, INy, INz) of the XY-axis acceleration sensor 31 and Z-axis contact switch 32 and impact input flag (FS). Specifically, stored is data concerning utilization ways, correction ratio, special correction conditions and special correction numbers. The tables are stored in plurality, including a recommended-position-set process table , a course process table, player-character moving table and an NPC moving table.

[0039]

The recommended-position set process table of Figure 25 is referred in a recommended-position set process hereinafter described with reference to Figure 38. Due to this table, the output value (INx, INy) of the XY-axis acceleration sensor 31 is utilized to determine a coordinate (Sx, Sy) of a sight. The output value (INz) of the Z-axis contact switch 32 and the impact input flag (FS) are not used.

The game map select processing table shown in Figure 26 is to be made reference to in a game map select process hereinafter described with reference to Figure 39. The output values (INx, INy) of the XY-axis acceleration sensor by this table are utilized for calculating a camera coordinate (Cx, Cy) change amount. Incidentally, because the correction ratio is wise, the camera coordinate (Cx,. Cy) will be moved twice the output value (INx, INy) of the XY-axis acceleration sensor 31. The output value (INz) of the Z-axis contact switch 32 is utilized for a map determining process. The impact input flag

(FS) is not utilized.

[0040]

The player character moving table shown in Figure 27 to Figure 30 is made reference to in a tilt movement process (Figure 44) to be executed at step 33, and in an impact movement process (Figure 45) to be executed in step 33 in a player character moving process hereinafter described with reference to Figure 42. The player character moving table includes tables for in-air, on-floor, on-ice and under-water, Any one of the conversion tables is to be selected and referred to in accordance with a coordinate topology where the player character is present (current position status).

[0041]

In the player character moving table, the output value X (INx) of the XY-axis acceleration sensor 31 is utilized for a change amount (dAx) of an X-movement acceleration while the output value Y (INy) is utilized for a change amount (dAy) of an Y-movement acceleration. In the case the current position status is "in-air", the moving-acceleration change amount (dAx, dAy) is zero by referring to Figure 27. For the case of "on-floor", because the correction ratio if referred to Figure 28 is twice, twice the output value (INx, INy) of the XY-axis acceleration sensor 31 gives a change amount (dAx, dAy) of moving acceleration. Also, where the output value (INx, INy) of the XY-axis acceleration sensor is greater than 20 due to particular correction condition 1, the moving-acceleration change amount (dAx, dAy) is limited to "40". For "on-ice", by referring to Figure 29 three times the output value (INx, INy) of the XY-axis acceleration sensor 31 gives a change amount (dAx, dAy) (greater moving amount "on-ice"). Meanwhile, where the output value (INx, INy) of the XY-axis acceleration sensor is greater than "20" due to particular correction condition 1, the moving-acceleration change amount (dAx, dAy) is limited to "60". For "under-water", by referring to Figure 30 a half

of the output value (INx, INy) of the XY-axis acceleration sensor 31 gives a moving-acceleration change amount (dAx, dAy) (smaller moving amount "in water"). Also, where the output value (INx, INy) of the acceleration sensor 31 is greater than "10" due to particular correction condition 1, the change amount (dAx, dAy) is limited to "5".

[0042]

In the player character moving tables, the output value (INz) of the Z-axis contact switch 32 is utilized for a change amount (dAz) of Z-movement acceleration of the player character where the output value of the Z-axis contact switch is 1 regardless of current position status, the change amount (dAz) in Z-movement acceleration is 1. Where the player character is in air and the Z-axis contact switch output value is 0, the change amount (dAz) in Z-movement acceleration is -1. .

There is no special correction condition.

[0043]

In the player-character moving table, an impact input flag (FS) has an effect upon X and Y moving-acceleration change amounts (dAx, dAy). In the case the present position status is "in-air" and "under-water", the impact input flag (FS) is ignored by referring to Figure 27 and Figure 30. Where the present position status is "on-floor", with reference to Fig. 28 processing is made to multiply by 3 times the X and Y moving-acceleration change amounts (dAx, dAy). Where the current position status is "on-ice", with reference to Figure 29 processing is made to multiply by 5 times the X and Y moving-acceleration change amounts (dAx, dAy). In this manner, when there is an impact input, for "on-floor" and "on-ice" the X and Y moving-acceleration change amounts (dAx, dAy) are increased (moved at higher speed) as compared to the usual.

[0044]

The NPC moving tables of Figure 31 and Figure 32 are to be referred to in a tilt

movement process (Figure 44) in step 44 and impact moving process (Figure 45) in step 45 of an NPC moving process hereinafter described with reference to Figure 43. The NPC moving tables includes tables for normal and upside-down positions. Any one of the two conversion tables is selected and referred to depending upon a pose (normal or upside-down) of a tortoise as NPC.

[0045]

In the NPC moving table, an output value X (INx) of the XY-axis acceleration sensor 31 is utilized to calculate a change amount (dAx) of an NPC X movement acceleration while an output value Y (INy) is utilized to calculate a change amount (dAy) of a Y movement acceleration. For the "normal position", because with reference to Figure 31 the correction ratio is 1/2 times, 1/2 times an output value (INx, INy) of the XY-axis acceleration sensor 31 gives an X-and-Y moving-acceleration change amount (dAx, dAy). Also, where the output the values (INx, INy) of the XY-axis acceleration sensor 31 is smaller than 10 under special correction condition 1, the moving-acceleration change amount (dAx, dAy) is 0 (in the "normal position", with a small tilt the tortoise will brace its legs and not slide). Also, where the output (INx, INy) of the XY-axis acceleration sensor 31 is greater than 20 under special correction condition 2, the moving-acceleration change amount (dAx, dAy) is limited to 10. For the "upside-down position", with reference to Figure 32, 2 times an output value (INx, INy) of the XY-axis acceleration sensor 31 gives an X-and-Y moving-acceleration change amount (dAx, dAy) (moving amount greater as compared to "normal"). Also, where the output value (INx, INy) of the XY-axis acceleration sensor 31 is greater than 20 under special correction condition 1, the moving-acceleration change amount (dAx, dAy) is limited to 40.

In the NPC moving tables, the output value (INz) of the Z-axis contact switch 32 is utilized to determine tortoise inversion to a normal or inverted position. Each time the

output value of contact switch 32 becomes "1", the tortoise turns to a normal or inverted state in a repetitive manner. The impact input flag (FS) is not utilized for the NPC movement process.

[0046]

Figure 33 is a flowchart of a main routine. If a cartridge 30 is loaded onto the game machine main body 10 and the power switch of the game machine main body 10 is turned on, the CPU 21 starts to process the main routine of Figure 33. First, in step 11 it is determined whether it is a first starting or not, or whether a player requested for 0G setting (e.g. whether started while pressing the operation key 13b of Figure 1) or not. If not a first starting and there was no 0G set request, the process advances to step 13. Meanwhile, when a first starting or there was a 0G set request, a 0G set process hereinafter described with reference to Figure 34 is made in step 12 and then the process proceeds to step 13. In the step 13, the player determines whether to set the neutral position to an arbitrary inclination or not. When determined to set to an arbitrary inclination, the process proceeds to step 14 to make a neutral position set process hereinafter described with reference to Figure 35 and the process advances to step 17. Where determined in the step 13 not to set to an arbitrary inclination, the process proceeds to step 15 to perform a recommended position set process hereinafter described with reference to Figure 36 to Figure 38, and the process advances to step 17. Here, the neutral-position setting is meant to set a reference tilt of the portable game apparatus for playing a game. The recommended position setting is meant to set a neutral position based on data wherein the data is concerned with a proper neutral position in accordance with a game content (the recommended position sight target coordinate 34d of the program ROM 34) that have been previously memorized in a game program.

In step 17 a game map select process hereinafter described with reference to

Figure 39 is performed so that one of a plurality of game maps is selected by the player. After the step 17, the process advances to a main loop.

[0047]

The main loop is a process of from step 19 to step 29, which is repeatedly executed until game over or game clear is reached. In step 19, required data is written to the display RAM 25 based on a coordinate (X, Y, Z) and pose number (PN) of the character data 26g of the work RAM 26, object character data 34a of the program ROM 34 and map data 34b. Based on the data stored in the display RAM, a game scene is displayed on the LCD 12. In step 20 a sensor output read process hereinafter described with reference to Figure 40 is performed. The output values of the XY-axis acceleration sensor 31 and Z-axis contact switch 32 are read out through the sensor interface 33 and then corrected. After the step 20, in step 21 it is determined whether there was a neutral-position set request or not. If there was no request, the process advances to step 23 while if there was a request the process proceeds to step 22 to perform a neutral-position set process. After resetting a neutral position, the process returns to step 19. This means that one operation key (e.g. operation key 13e shown in Figure 1) is assigned to an exclusive operation key for neutral-position setting so that neutral-position setting can be made at any time by pressing the operation key 13e even during playing a game.

[0048]

In step 23 it is determined whether the impact input flag is ON or not. If the impact input flag is OFF, the process proceeds to step 26 while if ON the process advances to step 24 to determine whether the topology of current coordinate that the player character is present is (current position status) under-water or not. If not under-water is determined, the process advances to step 26 while if determined under-water, the process advances to step 25 to perform a wave producing process (display is as shown in

the middle portion in Figure 19). Specifically, processing is made to cause waves in a direction and with a magnitude depending on a resultant vector, wherein the resultant vector is given by a vector component in the X-axis direction taken of a sensor output value X (INx) and a vector component in the Y-axis direction is taken of a sensor output value Y (INy). The player can have a feeling as if the impact applied by him or her to the portable game apparatus was reflected in an environment (water) of the game space. After step 25, the process proceeds to step 26.

In the step 26 an each-character moving process hereinafter described with reference to Figure 41 to Figure 47 is performed thereby performing a process of moving the player character and NPC. After the step 27, a collision process hereinafter described with reference to Figure 48 is performed thereby performing a process of colliding the player character with NPC, etc. After the step 27, a scroll process hereinafter described with reference to Figure 51 is performed.

[0049]

Figure 34 shows a subroutine flowchart for a 0G set process. This subroutine performs a process to store as 0G position data to backup RAM 35 an output value of the XY-axis acceleration sensor 31 when the portable game apparatus (specifically, the LCD 12 display surface) is held horizontal.

In step 121 "POSITION HORIZONTAL TO GROUND AND PRESS OPERATION KEY" is displayed on the LCD 12, requesting the player to hold the portable game apparatus (specifically, the LCD 12 display surface) in a horizontal state. In step 122 an operation key input process is performed. In step 123, if depression of an operation key (e.g. operation key 13b of Figure 1) for determination is determined, it is then determined in step 124 whether the Z-axis contact switch 32 is ON or not. When the Z-axis contact switch is ON, an alert sound is generated in step 125 and the process

returns to step 121. This is because, where the Z-axis contact switch is ON, the LCD in its display surface is directed downward and the player is requested to perform setting again. In step 124, where the Z-axis contact switch is determined OFF, then in step 126 the output value of the XY-axis acceleration sensor 31 at this time is stored as 0G position data to the backup RAM 35.

[0050]

Figure 35 is a subroutine flowchart for a neutral-position set process. This subroutine performs process that the player arbitrarily determines a portable game apparatus at a holding angle easy to play a game. The output value of the XY-axis acceleration sensor 31 and Z-axis contact switch 32 at that time are stored as neutral position data to the work RAM 26.

In step 141 "POSITION AT ANGLE EASY TO PLAY AND PRESS OPERATION KEY" is displayed on the LCD 12. In step 142 an operation key input process is made. In step 143, if the depression of an operation key for determination (e.g. operation key 13b of Figure 1) is determined, then in step 144 correction is performed by subtracting 0G position data from an output value of the XY-axis acceleration sensor 31 at this time (the neutral position data is rendered as data corresponding to a tilt with respect to the horizontal state). Then, in step 145 a correction value of the output of the XY-axis acceleration sensor (calculation result of step 144) and an output value of the Z-axis contact switch 32 are stored as neutral position data to the neutral position data memory area 26a of the work RAM 26.

[0051]

Figure 36 and Figure 37 are an example of the LCD screen of a recommended-position set process. On the LCD 12, a sight target coordinate 71 is fixingly displayed based on a recommended-position sight target coordinate stored in the program ROM 34.

Furthermore, a sight 72 is displayed to move in display according to a tilt of the portable game apparatus. The player moves the sight 72 by tilting the portable game apparatus. The tilt of the portable game apparatus when the sight 72 overlaps with the sight target coordinate 71 (Figure 37) is set as recommended position.

[0052]

Figure 38 is a flowchart of a recommended-position set process. First, in step 151, "MOVE SIGHT TO TARGET COORDINATE AND PRESS OPERATION KEY" is displayed on the LCD 12. In step 152 read out is an output value (INx, INy, INz) of the XY-axis acceleration sensor 31 and Z-axis contact switch 32. In step 153 after the step 152, reference is made to a table for recommended-position set process shown in Figure 25. Referring to this table, a sight coordinate (Sx, Sy) is determined based on the sensor output value (INx, INy) of the XY-axis acceleration sensor 31. In step 154, a sight 72 is displayed based on a determined sight coordinate (Sx, Sy) on the LCD 12. For example, where the portable game apparatus is tilted in a plus direction about the X-axis, the sight is displayed in a lower region with respect to a center of the display surface of the LCD 12. As the inclination is greater, display is in a position closer to a lower end of the display surface. Where the tilt is in the minus direction about the X-axis, the sight is displayed in the upper region with respect to the center of the LCD 12 display surface. As the inclination is greater, display is in a position closer to the upper end of the display surface. Also, where tilt is in the plus direction about the Y-axis, display is in the right region while where tilt is in the minus direction, display is in the left region. The player is allowed to control the sight 72 in movement-display in this manner by tilting the portable game apparatus to move it onto the sight target coordinate 71 displayed in a fixing fashion (Figure 37).

[0053]

In the step 155 after the step 154, an operation-key input process is made. In step 156 after the step 155, it is determined whether an operation key for determination (e.g. operation key 13b in Figure 1) was pressed or not. When the determination key was not pressed is determined, the process returns to the step 152. The determination key was pressed is determined, it is determined in step 157 whether the sight 72 and the sight target coordinate 71 are in overlap or not. If determined not overlapped, in step 162 alert sound is generated and the process returns to step 151 to request the player to set again a recommended position. If it is determined in step 157 that the sight 72 and the sight target coordinate 71 are in overlap, then in step 158 it is determined whether the Z-axis contact switch 32 is ON or not. If it is determined that the Z-axis contact switch 32 is ON, the process advances to step 162 to generate alert sound and the process returns to the step 151 to request the player to set again recommended position. In step 158, if it is determined that the Z-axis contact switch is off, the process proceeds to step 159 to read out an output value of the XY-axis acceleration sensor 31 and subtract 0G position data from the read-out value. In step 161, an output correction value of the XY-axis acceleration sensor (calculation result in step 159) and Z-axis contact switch output value is stored to the neutral position data area 26a of the work RAM 26.

[0054]

Figure 39 is a flowchart of a game map select process. In this subroutine, the player selects any one of a plurality of game maps stored in the game program. The screen of game map select process is displayed, for example, as shown in Figure 16 mentioned before. On the LCD 12, one area of a game-map select map is displayed. The player makes slide input in the X-axis or Y-axis direction to move the display area on the LCD 12 thereby displaying map icons (A, B, C, D in Figure 15) within the display area. Then, an acceleration is inputted in the Z-axis direction. This results in selection of a

game course corresponding to a course icon being displayed on the LCD 12 upon inputting the acceleration in the Z-axis direction.

[0055]

First, in step 171 a camera coordinate (Cx, Cy) is initialized. Then, in step 172 one area of the game-map select map is displayed on the LCD 12 based on the camera coordinate (Cx, Cy). In step 173 a sensor output read process hereinafter described with referring to Figure 40 is made. As a result, the output values of the XY-axis acceleration sensor 31 and Y-axis contact switch 32 are read out and corrected. In step 174 a table shown in Figure 26 is referred to. Specifically, the camera coordinate (Cx, Cy) is changed based on the sensor output values (INx, INy). Because the correction ratio is twice, the camera coordinate (Cx, Cy) is varied by an amount twice the sensor output value (INx, INy). For example, when the sensor output value (INx) is 5, the camera coordinate (Cx) is rendered +10. In step 175 it is determined whether the display area based on the camera ordinate (Cx, Cy) is outside a range of the game map select map or not. If not outside the range, the process advances to step 177 while if in outside the range the process proceeds to step 176. In step 176 correction is made so as to display an end area of the game-map select map and then the process proceeds to step 177. In the step 177 it is determined whether the Z-axis contact switch 32 is ON or not. If the contact switch 32 is determined OFF, the process returns to step 172. If the Z-axis contact switch 32 is determined ON, then it is determined in step 178 whether any one of the map icons (A, B, C, D in Figure 15) is displayed in the display range of the LCD 12 or not. If it is determined that no map icon is displayed within the display area, then in step 179 an alert sound is generated and the process returned to step 172. If it is determined that a map icon is displayed within the display range, then in step 181 a corresponding game map number (MN) to the map icon being displayed is stored to the work RAM 26.

[0056]

Figure 40 is a flowchart for a sensor output read process. In this subroutine, the output values of the XY-axis acceleration sensor 31 and Z-axis contact switch 32 are read out and corrected. Specifically, from the data of the latch 334 and latch 335 of the sensor interface 33 are read output values (INx, INy) of the acceleration sensor and an output value (INz) of the Z-axis contact switch 32. Furthermore, a correction process is made based on 0G position data and neutral position data.

[0057]

In step 201, data is read out of the latch 334 and latch 335. In step 202, acceleration-sensor output values (INx, INy) and Z-axis contact switch output value (INz) are read from the latch data, and stored to the acceleration-sensor output value memory area 26b of the work RAM 26. In step 203 it is determined whether there was an impact input or not. Specifically, it is determined whether equal to or greater than a given value a magnitude of a resultant vector having vector component in the X-axis direction taken of the acceleration sensor 31 output value X (INx) and a vector component in the Y-axis direction taken of the acceleration sensor 31 output value Y (INy). If determined equal to or greater than a given value, then in step 204 the impact input flag (FS) is set "ON" and the process advances to step 206. If the resultant vector magnitude is determined smaller than the given value, then in step 205 the impact input flag (FS) is set "OFF" and the process advances to step 206. In step 202, processing is made to subtract the 0G position data memorized in the backup RAM 35 from the data of the acceleration-sensor output value memory area 26b. In step 207, the value further corrected with the neutral position data is stored as INx, INy and INz to the acceleration-sensor output memory area 26b.

The correction with the neutral position data is performed, specifically, on the

output value X (INx) and output value Y (INy) of the acceleration sensor by subtracting the values of the neutral position data (NPx, NPy). For the output value (INz) of the Z-axis contact switch 32, when the value of neutral position data (NPz) is "1", processing is made to invert "0" and "1".

[0058]

Figure 41 to Figure 47 are flowcharts for an object moving process. Figure 41 is an object moving process main routine flowchart. In step 261, a player-character moving process is performed that is hereinafter described with reference to Figure 42. In step 262, an NPC moving process is performed that is hereinafter described with reference to Figure 43. The NPC moving process is repeated the number of NPCs.

[0059]

Figure 42 is a player-character moving process flowchart. In step 31, a present coordinate (X, Y, Z) of the player character is stored by copy as a last-time coordinate (Px, Py, Pz). This is required to return the player character collided with a wall to a last-time coordinate, in a collision process hereinafter described with reference to Figure 48. In step 32, a moving-acceleration change amount (dAx, dAy, dAz) is initialized. In step 33, a tilt movement process is performed that is hereinafter described with reference to Figure 44. In step 34, an impact moving process is performed that is hereinafter described with reference to Figure 45. In step 35, a jump moving process is made that is hereinafter described with reference to Figure 46. It is determined in step 36 whether a wave generation process in step 25 of the flowchart of Figure 33 has been made or not. If no wave generation is determined, the process advances to step 38. If waves have generated is determined, in step 37 a wave moving process hereinafter described with reference to Figure 47 is made, and then the process proceeds to step 38. In the step 38, a moving acceleration (Ax, Ay, Az) is calculated based on the moving-acceleration change amount

(dAx, dAy, dAz) calculated in the tilt moving process, impact moving process, jump process and wave moving process of the step S33 to S37, and a velocity (Vx, Vy, Vz) is calculated based on the moving acceleration (Ax, Ay, Az). In step 39, a coordinate (X, Y, Z) is calculated based on the velocity (Vx, Vy, Vz).

[0060]

Figure 43 is a flowchart of an NPC movement process. In step 41 a current coordinate (X, Y, Z) is stored by copy to the last-time coordinate (Px, Py, Pz). In step 42 the moving-acceleration change amount (dAx, dAy, dAz) are initialized. In step 43 an NPC self-controlled movement process is executed based on the game program. Specifically, a moving-acceleration change amount (dAx, dAy, dAz) e.g. for a tortoise is determined based on a random number value. In step 44, a tilt movement process is executed that is hereinafter described with reference to Figure 44. In step 45 an impact process is made that is hereinafter described is the reference to Figure 45. In step 46 it is determined whether a wave producing process has been made in step 25 of the flowchart of Figure 33 or not. If no wave production is determined, the process advances to step 48. If waves have been produced is determined, then in step 47 a wave movement process hereinafter described with reference to Figure 46 is executed and then the process advances to step 48. In step 48, a moving acceleration (Ax, Ay, Az) is calculated based on the moving-acceleration change amounts (dAx, dAy, dAz) determined by the self-controlled movement process, tilt movement process, impact movement process and wave movement process of step 43 to 47. Furthermore, a velocity (Vx, Vy, Vz) is calculated based on the movement acceleration (Ax, Ay, Az). In step 49 a coordinate position (X, Y, Z) is calculated based on the velocity (Vx, Vy, Vz). In step 51 it is determined whether an output value (INz) of the Z-axis contact switch is "1" or not. In the case that the Z-axis contact switch output value (INz) is "0", the NPC movement

process subroutine is ended. Where the Z-axis contact switch output value (INz) is "1", an inversion process to a normal or upside-down position is executed in step 52. Specifically, a pose number (PN) of the character data in the work RAM 26 is changed.

[0061]

Figure 44 is a flowchart for an inclination movement process. In this subroutine, processing is made to calculate a movement-acceleration change amount (dAx, dAy) such that the character (player character and NPC) rolls (slides) in accordance with a tilt (tilt input) of the portable game apparatus. In step 331, an acceleration-sensor-output conversion table is selected. Specifically, in the case of movement process of a player character, any one of "in-air", "on-floor", "on-ice" and "under-water" in Figure 30 is selected according to a current positional status. For an NPC movement process, any one of "for normal position", and "for upside-down position" in Figure 31 or Figure 32 is selected according to a pose number in step 323 after the step 321, reference is made to the selected conversion table to calculate an X-movement-acceleration change amount (dAx) and Y-movement-acceleration change amount (dAy) is calculated from a sensor-output value X (INx) and sensor output value Y (INy).

[0062]

Figure 45 is a flowchart of an impact movement process. In this subroutine, processing is made to increase the movement-acceleration change amount (dAx, dAy) such that the player character dashes (moves at high speed) when an impact is given. Incidentally, because an impact input flag is set neglectedly in the NPC movement tables (Figure 31 and Figure 32), there is no change in NPC movement acceleration change amount due to impact input. However, the NPC may be set to move at high speed where there is an impact input. In step 341, an acceleration-sensor output conversion table is selected. Specifically, any one of "in-air", "on-floor", "on-ice" and "under-water" in

Figure 27 to Figure 30 is selected according to a current positional status. In step 342 reference is made to selected conversion table to increase an X-movement-acceleration change amount (dAx) and Y movement-acceleration change amount (dAy) based on value of the impact input flag (FS).

[0063]

Figure 46 shows a flowchart of a jump process. In this subroutine, when there is an acceleration input in the Z-axis direction when the Z-axis contact switch output value (INz) is 1, processing is made to cause the player character to jump. Also, when there is no movement input in the Z-axis direction in a state the player character is in the air, processing is made to descend the player character.

In step 351 it is determined whether the output value (INz) of the Z-axis contact switch 32 is 1 or not. When the output value (INz) of contact switch 32 is "1", the current position status (PS) is set as "in-air" in step 52. Thereafter in step 353 the Z moving-acceleration change amount (dAz) is rendered "1". When the output value (INz) of the Z-axis contact switch 32 is "0" in the step 351, it is determined in step 354 whether the player character is "in-air" or not. When not "in-air", the jump process is ended. Where "in-air" in the step 354, the Z moving-acceleration change amount (dAz) is rendered "−1" in step 355 and then the jump process is ended.

[0064]

Figure 47 shows a flowchart of a wave movement process. In this subroutine, processing is made to calculate a moving-acceleration change amount due to the waves produced due to impact input by the player. In step 361 a current position status is read in. In step 362 it is determined whether the current position status is in a position to undergo an affection of waves or not (i.e. "under-water" or not). If determined as a position free from an affection of waves, the wave movement process is ended. If determined as a

position to undergo an affection of waves, then in step 363 are calculated respective X and Y moving-acceleration change amounts due to an affection of waves and added to the X and Y moving-acceleration change amounts calculated by the tilt movement process and impact movement process.

[0065]

Figure 48 shows a flowchart of a collision process. In step S271 to S275, an NPC collision determination process is carried out. The NPC collision determination process is repeated to the number of NPCs. In step 271 it is determined whether an NPC has collided with a wall or not. If determined as collision with a wall, the process proceeds to step 273. If no collision is determined, the process advances to step 272 wherein it is determined whether there has been a collision with another NPC or not. If determined as collision with another NPC, the process advances to step 272. If determined as no collision with another NPC, the process proceeds to step 273. Where determined as a collision with a wall or another NPC, then in step 273 an impact sound is generated and then in step 274 the NPC coordinate (X, Y, Z) is returned to the last-time coordinate (Px, Py, Pz), and the process advances to the step 275.

In step 275, a current position status of NPC is detected and stored in the work RAM 26. After step 275 it is determined in step 276 whether the player character has collided with a wall or not. If no collision against wall is determined, the process proceeds to step 279. If a collision with a wall is determined, then in step 277 an impact sound is generated and then in step 278 the player character coordinate (X, Y, Z) is returned to the last-time coordinate (Px, Py, Pz), and the process advances to step 279.

In step 279, a current position status of the player character is detected and stored in the work RAM 26. After step 279, it is determined in step 281 whether the player character has collided with an NPC or not. If a collision against an NPC is determined, a

process is made in step 282 to vanish the NPC. After step 282, it is determined in step 283 whether all the NPCs have been vanished or not. If all the NPCs have vanished is determined, a game clear process is executed in step 284. When no collision with an NPC is determined in step 281 or when all the NPCs have not vanished is determined in step 283, the process proceeds to step 285. In step 285 it is determined whether the player character has fallen in a hole or not. If determined fallen in a hole, a game over process is effected in step 286. Where the determination is not fallen in a hole, the impact process is ended.

[0066]

Figure 49 and 50 each show one example of a scene showing on-screen scroll. In the scene, there are displayed a ball as a player character, tortoises 62a – 62c as NPC, and a wall 63 and hole 64 forming a maze. The dotted lines 65 show a limit of screen scroll (actually, the dotted lines 65 will not be displayed on the LCD 12). The game map is a virtual map that is broader than LCD 12 display area, as stated before. On the LCD 12 is displayed part of a game map around the player character 61. When the player tilts or so the portable game apparatus and the player character 61 is moving to an outer area of the dotted lines 65, the scene is scrolled moving the game-map display area over the LCD12. Furthermore, the player character 61 and NPC 62 are moved to and displayed in a position toward a center of a scene by a corresponding amount to scrolling. In this manner, screen scrolling makes possible game play with a broader game map. For example, if the player character is going beyond the dotted line 65 to a left side area as shown in Figure 49, the game map area in display is scrolled to left so that the player character 61 and NPC can be moved to and displayed in a position by a corresponding amount to scrolling (Figure 50). Note that the scroll rate may be changed depending upon a magnitude of tilt input.

[0067]

Figure 51 shows a flowchart of a screen scroll process. In step 291 it is determined whether the player character is out of a scroll area in an X-axis minus direction or not. Here, the scroll area refers to an area as surrounded by the dotted lines 65 shown in Figure 38. If determined not out of the area with respect to the X-axis minus direction, the process advances to step 294. If determined out of the area in the X-axis minus direction, it is then determined in step 292 whether the current display area on the LCD 12 is a left end area of the game map or not. If determined as a left end area, the process advances to step 294. If determined not a left end area, then in step 293 a scroll counter X coordinate (SCx) stored in the display RAM 25 is decreased by a given amount and then the process proceeds to step 294. In step 294 it is determined whether the player character is out of the scroll area with respect to the X-axis plus direction or not. When determined not out of the area in the X-axis plus direction, the process advances to step 297. When determined out of the area in the X-axis plus direction, it is determined in step 295 whether the current display area on the LCD 12 is a right end area of the game map or not. If determined as a right end area, the process advances to step 297. When determined not a right end area, in step 296 the scroll counter X coordinate (SCx) is increased by a given amount and then the process proceeds to step 297.

[0068]

In step 297 it is determined whether the player character is out of the scroll area in a Y-axis minus direction or not. If determined not out of the area in the Y-axis minus direction, the process advances to step 301. When determined out of the area in the Y-axis minus direction, it is determined in step 298 whether the current display area on the LCD 12 is an upper end area of the game map or not. If determined as an upper end area, the process proceeds to step 301. When determined not an upper end area, in step 299 a

scroll counter Y coordinate (SCy) is decreased by a given amount and then the process proceeds to step 301. In step 301 it is determined whether the player character is out of the scroll area in a Y-axis plus direction or not. When determined not out of the area in the Y-axis plus direction, the screen scroll process is ended. When determined out of the area in the Y-axis plus direction, it is determined in step 302 whether the current display area on the LCD 12 is an lower end area of the game map. When determined as a lower end area, the screen scroll process is ended. When determined not a lower end area, in step 303 the scroll counter Y coordinate (SCy) is decreased by a given amount and then the screen scroll process is ended.

[0069]

Although the above embodiment is for one player to enjoy a game, a plurality of portable game apparatuses may be used through communication cables or wireless communication to enable a plurality of players to enjoy games. In such a case, the game characters or game world (game map) may be shared so that a change in a game character or game world based on one player's operation is reflected on the portable game apparatuses for other players. For example,. it is possible to contemplate a game one player tilts the portable game apparatus to tilt the game world (game map) while another player tilts the portable game apparatus to manipulate on a moving direction of a game character (ball) thereby moving the ball in the game map. Meanwhile, it is also possible to consider a game that the role is shared between players to control one character. For example, one player tilts the portable game apparatus to movement-control a game character (ball) while another player gives an acceleration input in the Z-axis direction to the portable game apparatus, thereby causing the game character (ball) to jump so that a game course can be cleared by cooperative operation by the both.

[0070]

Although in the above embodiment the control on the game character is based solely on an output of the acceleration sensor, the game character or the like may be controlled by a combination of outputs of the operation key and the acceleration sensor. For example, in a pinball game, it is possible to contemplate such a game that the ball is moved and controlled by inclining or swinging the portable game apparatus wherein a flipper is operated upon pressing the operation key. Also, in a game represented by TETRIS (registered trademark) game wherein an object falls from the above, it is possible to consider such a game that the object is controlled in movement by inclining or swinging the portable game apparatus wherein the object is changed in direction by the operation key or moved at high speed by giving an impact input or deformed by applying an acceleration input in the Z-axis direction.

[0071]

Although in the above embodiment the acceleration sensor was provided on the cartridge, the acceleration sensor may be provided on the side of the game apparatus main body. In the case of providing an acceleration sensor on the side of the game apparatus main body, there is no need to provide an acceleration sensor for each cartridge, reducing cost. Also, the information storage medium used for the portable game apparatus is not limited to a cartridge but may be an IC card, such as a PC card.

[0072]

Although in the above embodiment the neutral position data was stored on the work RAM 26 and set up each time of game play, it may be stored on the backup RAM 35 so that the same data can be utilized in next-round of game play.

[0073]

Although in the above embodiment the neutral position was determined by a player, neutral position data may be previously stored in a game program so that it can be

utilized. Also, a plurality of neutral position data may be stored so that a player can select any of them.

[0074]

In the above embodiment, although the acceleration sensor was to detect a tilt of the portable game apparatus as concerned with movement control of the game character, the movement of the portable game apparatus may be detected by the acceleration sensor.

[0075]

In the first embodiment, the game characters employed only the player character (ball) and enemy character (tortoise). However, in addition to them, it is possible to appear NPC (non-player character), such as ally characters, assisting the player character or neutral characters. These NPCs, although self-controlled according to a game program (NPC not self-controlled may be provided), may be moved or deformed according to an operation (tilt, input or impact input) by a player.

[Brief description of the drawings]

[Figure 1]

Figure 1 is an external view of a portable game apparatus of one embodiment of the present invention.

[Figure 2]

Figure 2 is a view showing a definition of XYZ axes.

[Figure 3]

Figure 3 is a block diagram of the portable game apparatus of one embodiment of the invention.

[Figure 4]

Figure 4 is a block diagram of a sensor interface.

[Figure 5]

Figure 5 is a diagram showing a principle on measuring the output of an acceleration sensor.

[Figure 6]

Figure 6 is a view showing a structure of a Z-axis contact switch.

[Figure 7]

Figure 7 is a view showing that an acceleration input in the Z-axis direction is detected by the Z-axis contact switch.

[Figure 8]

Figure 8 is a view showing a game scene of an embodiment of the invention.

[Figure 9]

Figure 9 is an illustrative view showing a slide input.

[Figure 10]

Figure 10 is a view showing a tilt input about the X-axis as a center.

[Figure 11]

Figure 11 is a view showing a Y-axis slide input.

[Figure 12]

Figure 12 is a view showing a tilt input about the Y-axis as a center.

[Figure 13]

Figure 13 is a view showing an impact input in the X-axis direction.

[Figure 14]

Figure 14 is a view showing an impact input in the Y-axis direction.

[Figure 15]

Figure 15 is a view showing an acceleration input in the Z-axis direction.

[Figure 16]

Figure 16 is a view showing a utilization method of a slide input.

[Figure 17]

Figure 17 is a view showing a utilization method of a tilt input about the X-axis as a center.

[Figure 18]

Figure 18 is a view showing a utilization method of a tilt input about the Y-axis as a center.

[Figure 19]

Figure 19 is a view showing a utilization method of an impact input.

[Figure 20]

Figure 20 is a view showing a utilization method of an acceleration input in the Z-axis direction.

[Figure 21]

Figure 21 is a memory map of a program ROM.

[Figure 22]

Figure 22 is a memory map of a work RAM.

[Figure 23]

Figure 23 is a memory map of a display RAM.

[Figure 24]

Figure 24 is a memory map of a backup RAM.

[Figure 25]

Figure 25 is an acceleration-sensor output conversion table (for recommended-position-set processing).

[Figure 26]

Figure 26 is an acceleration-sensor output conversion table (for game-map-select processing).

[Figure 27]

Figure 27 is an acceleration-sensor output conversion table (for player-character movement/in-air).

[Figure 28]

Figure 28 is an acceleration-sensor output conversion table (for player-character movement/on-floor).

[Figure 29]

Figure 29 is an acceleration-sensor output conversion table (for player-character movement/on-ice).

[Figure 30]

Figure 30 is an acceleration-sensor output conversion table (for player-character movement/under-water).

[Figure 31]

Figure 31 is an acceleration-sensor output conversion table (for NPC movement/face-up).

[Figure 32]

Figure 32 is an acceleration sensor output conversion table (for NPC movement/face-up).

[Figure 33]

Figure 33 is a main routine flowchart.

[Figure 34]

Figure 34 is a 0G set process flowchart.

[Figure 35]

Figure 35 is a neutral-position set process flowchart.

[Figure 36]

Figure 36 is an example of a recommended-position-set-process screen.

[Figure 37]

Figure 37 is a screen that a sight and a target coordinate are coincident in the recommended-position-set process.

[Figure 38]

Figure 38 is a recommended-position-set-process flowchart

[Figure 39]

Figure 39 is a game map select process flowchart.

[Figure 40]

Figure 40 is a sensor output read process flowchart.

[Figure 41]

Figure 41 is an each-object moving process flowchart.

[Figure 42]

Figure 42 is a player-character moving process flowchart.

[Figure 43]

Figure 43 is an NPC moving process flowchart.

[Figure 44]

Figure 44 is a tilt-moving process flowchart.

[Figure 45]

Figure 45 is an impact-moving process flowchart.

[Figure 46]

Figure 46 is a jump moving process flowchart.

[Figure 47]

Figure 47 is a wave moving process flowchart.

[Figure 48]

Figure 48 is a collision process flowchart.

[Figure 49]

Figure 49 is a screen-scroll explanatory view (before scroll).

[Figure 50]

Figure 50 is a screen-scroll explanatory view (after scroll).

[Figure 51]

Figure 51 is a screen-scroll process flowchart.

[Explanation of reference characters]

10 ··· main body of game machine

12 ··· LCD

13 ··· operation key

21 ··· CPU

25 ··· display RAM

26 ··· work RAM

30 ··· game cartridge

31 ··· XY-axis acceleration sensor

32 ··· Z-axis contact switch

33 ··· sensor interface

34 ··· program ROM

35 ··· backup RAM

[Document name] ABSTRACT

[Abstract]

[Problem]

Where providing an acceleration sensor to a portable game apparatus, the inclination of the portable player held by the player is not constant.

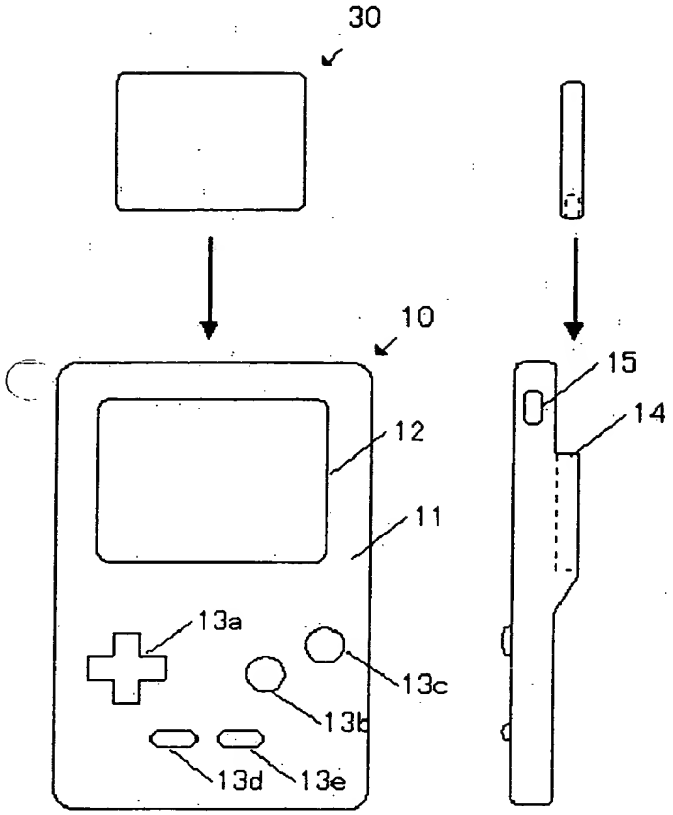
[Solving means]

Provided are neutral-position set means for setting as a neutral position a tilt of the portable game apparatus held by a player, temporary storage means for storing as neutral position data an output value of the acceleration sensor corresponding to the neutral position, and correction means for correcting the output value of the acceleration sensor based on the neutral position data. Game control means changes the display of game scene based on the output of the correction means.

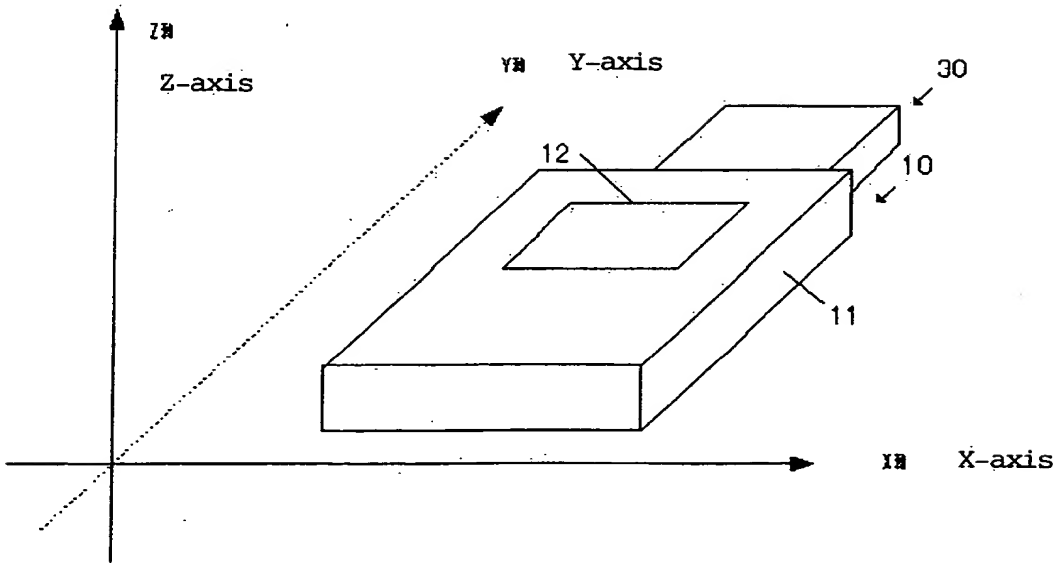
[Selected figure] Figure 33

【書類名】　図面　[Document Name] Drawings

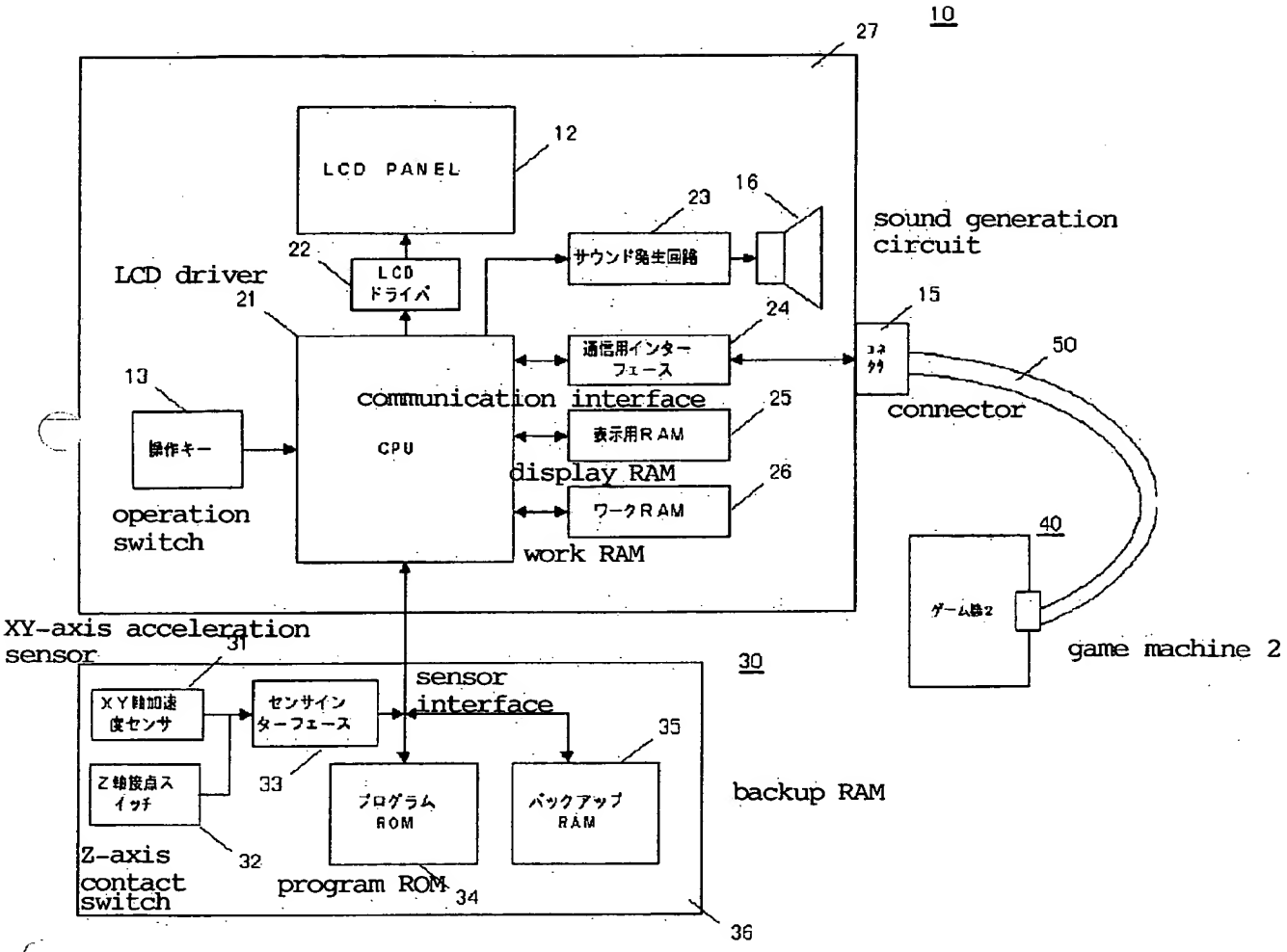
【図１】　[Figure 1]



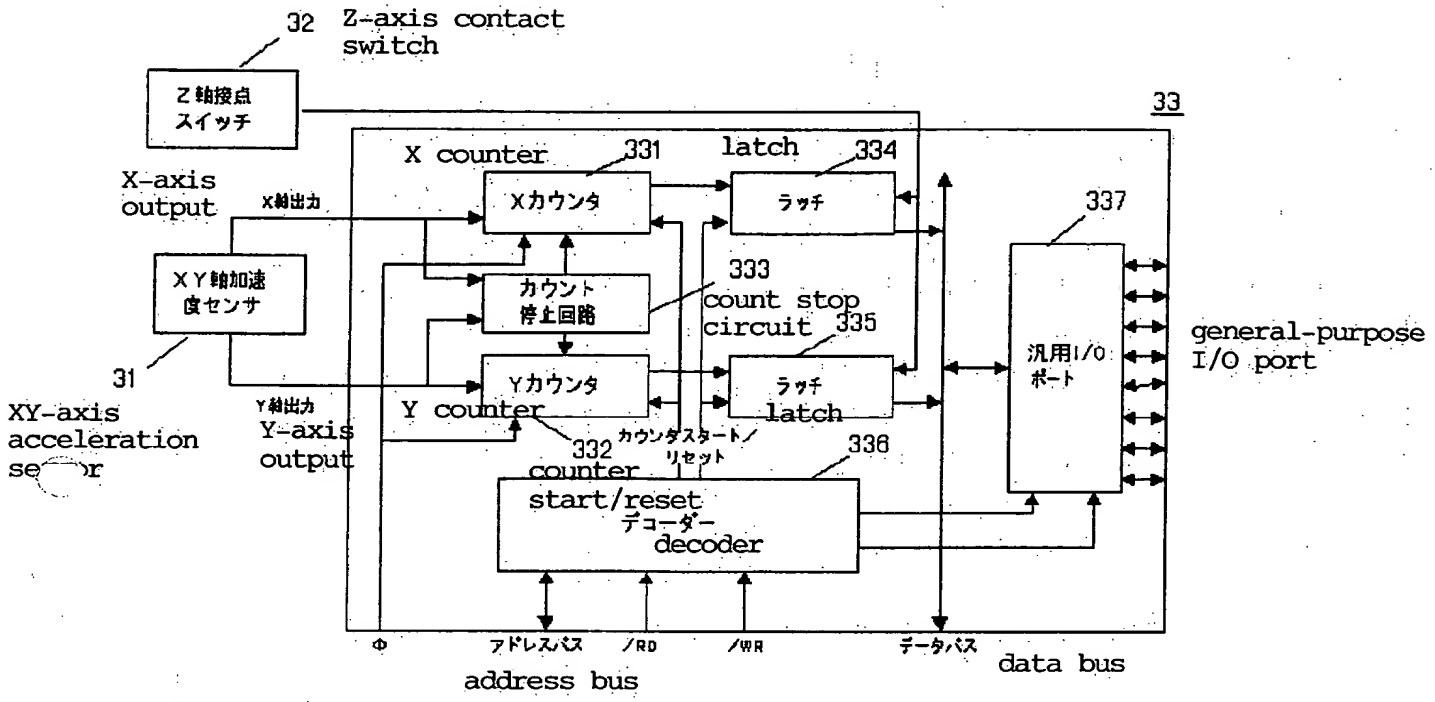
【図２】　[Figure 2]



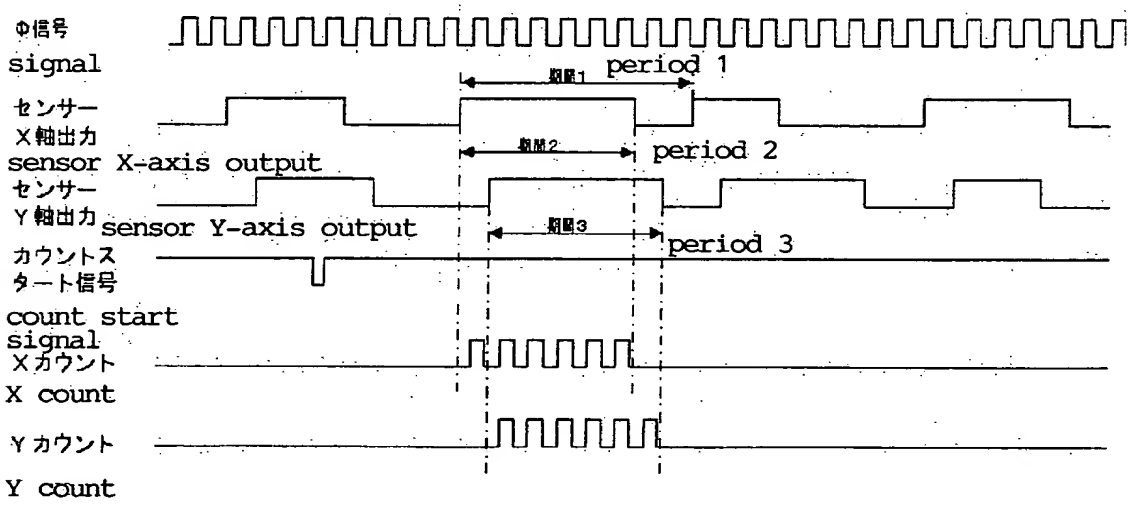
【図３】　[Figure 3]


10
27
LCD PANEL
12
16
23
sound generation circuit
サウンド発生回路
22
LCD ドライバ
LCD driver
21
15
24
通信用インターフェース
50
13
communication interface
connector
操作キー
CPU
25
表示用RAM
display RAM
26
ワークRAM
operation switch
work RAM
40
ゲーム機2
game machine 2
XY-axis acceleration sensor
31
30
XY軸加速度センサ
センサインターフェース
sensor interface
35
Z軸接点スイッチ
33
プログラムROM
バックアップRAM
backup RAM
Z-axis contact switch
32
program ROM
34
36

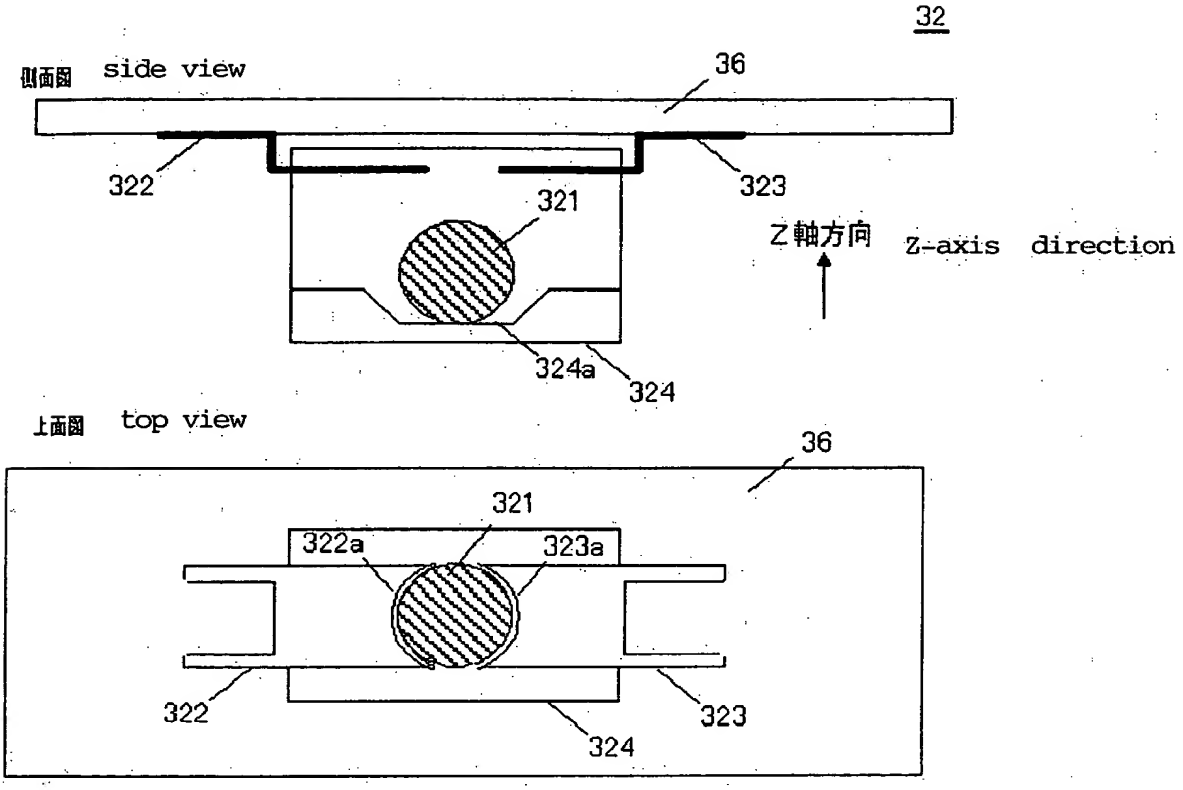
【図４】 [Figure 4]



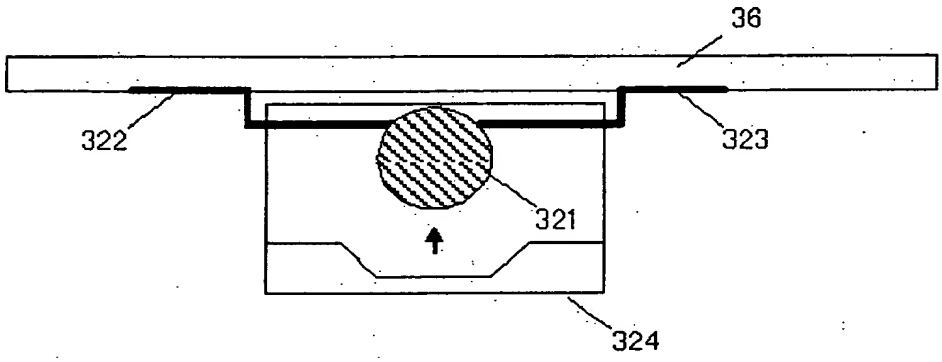
【図５】 [Figure 5]



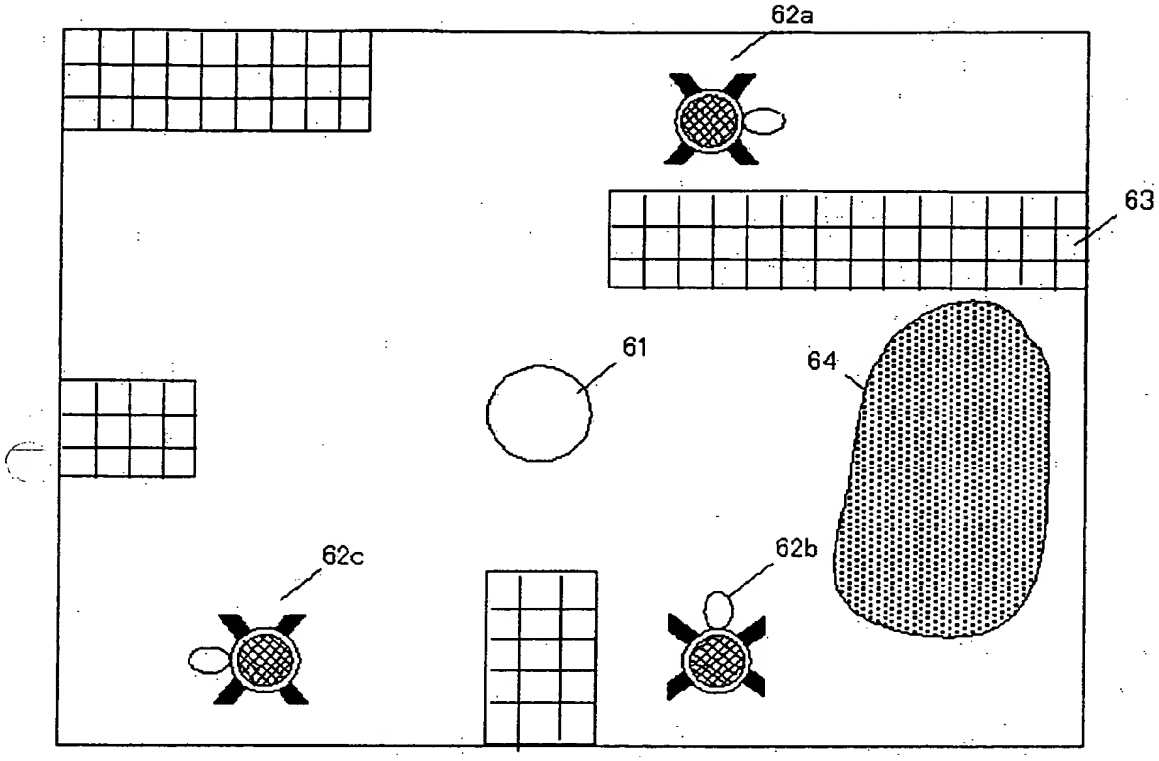
【図６】 [Figure 6]



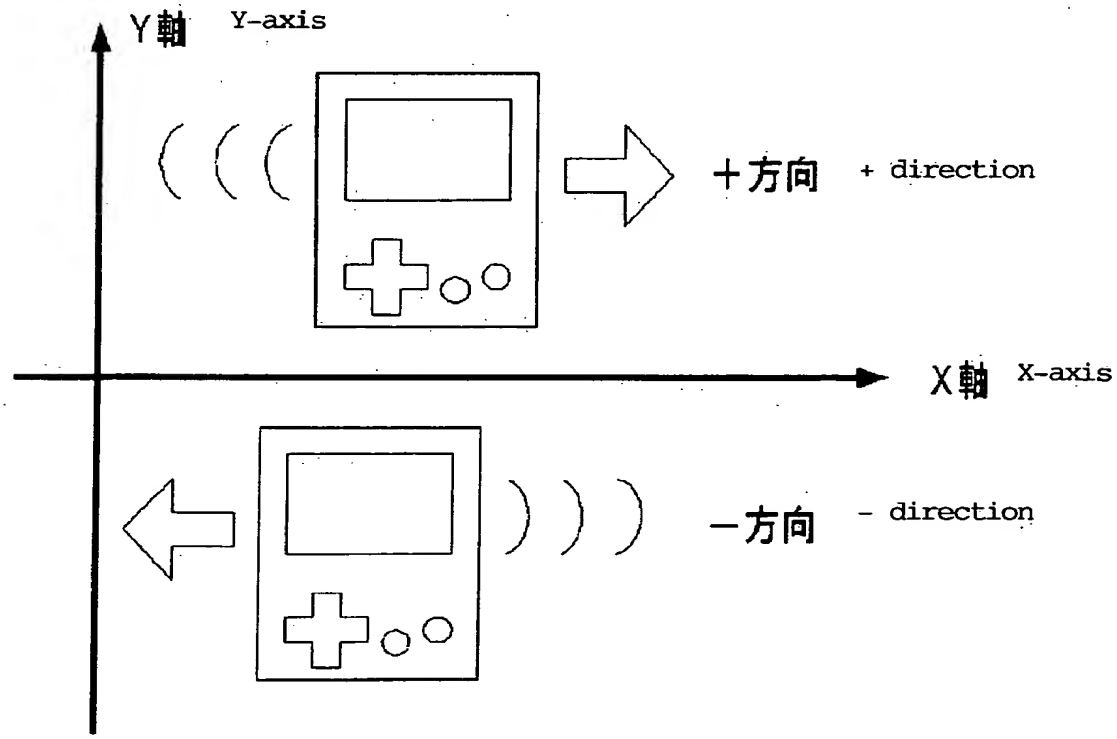
【図７】 [Figure 7]



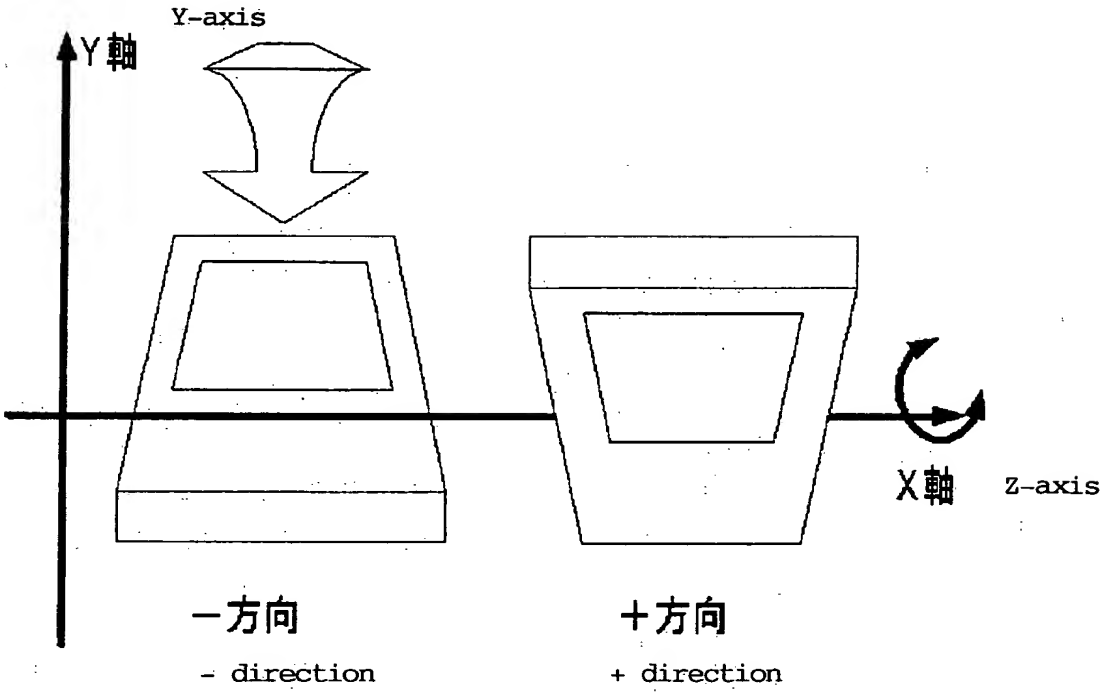
【図８】 [Figure 8]



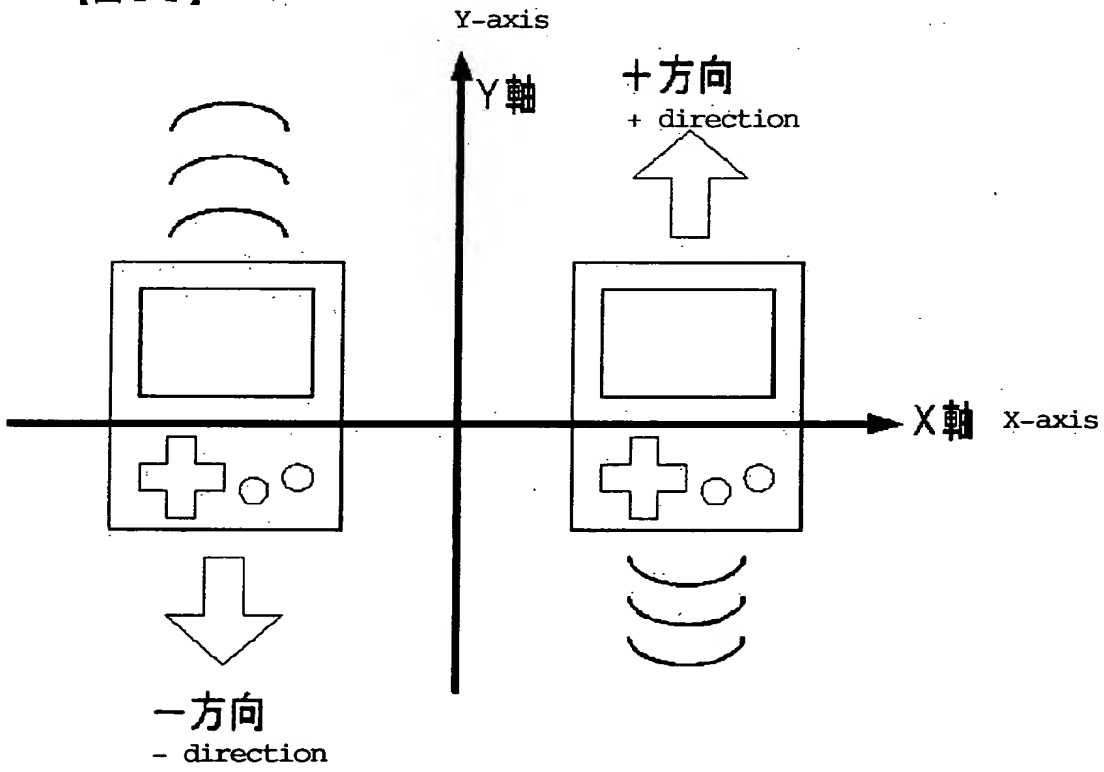
【図９】 [Figure 9]



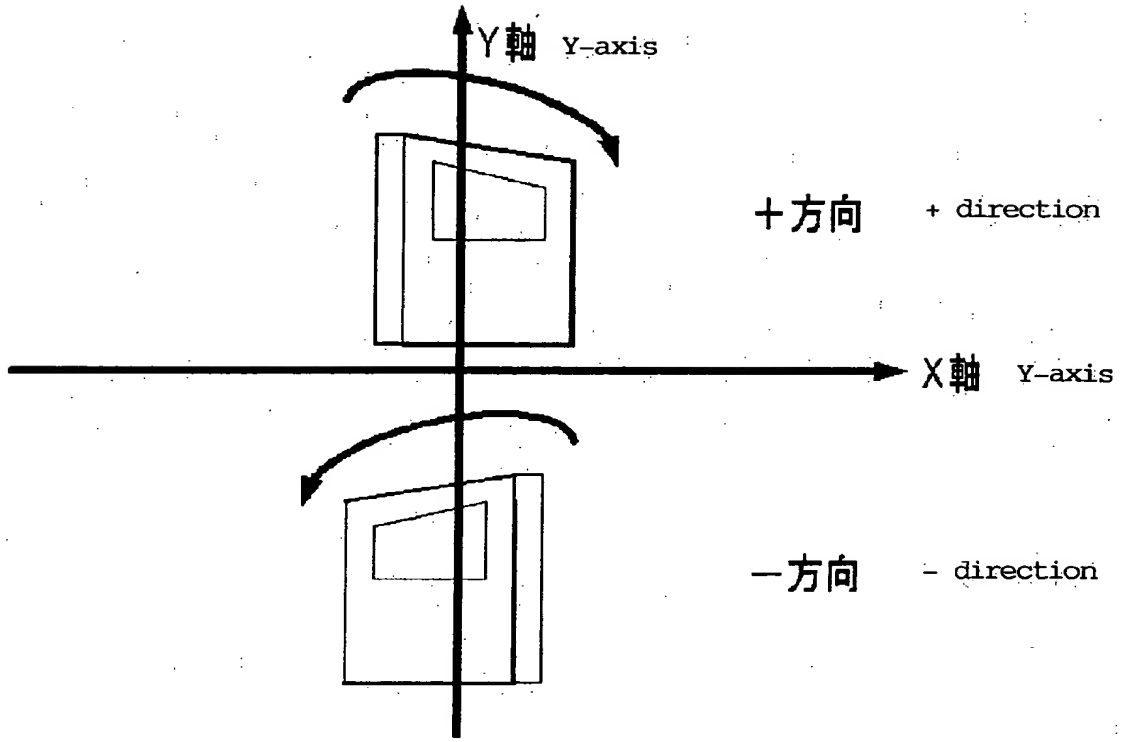
【図１０】 [Figure 10]



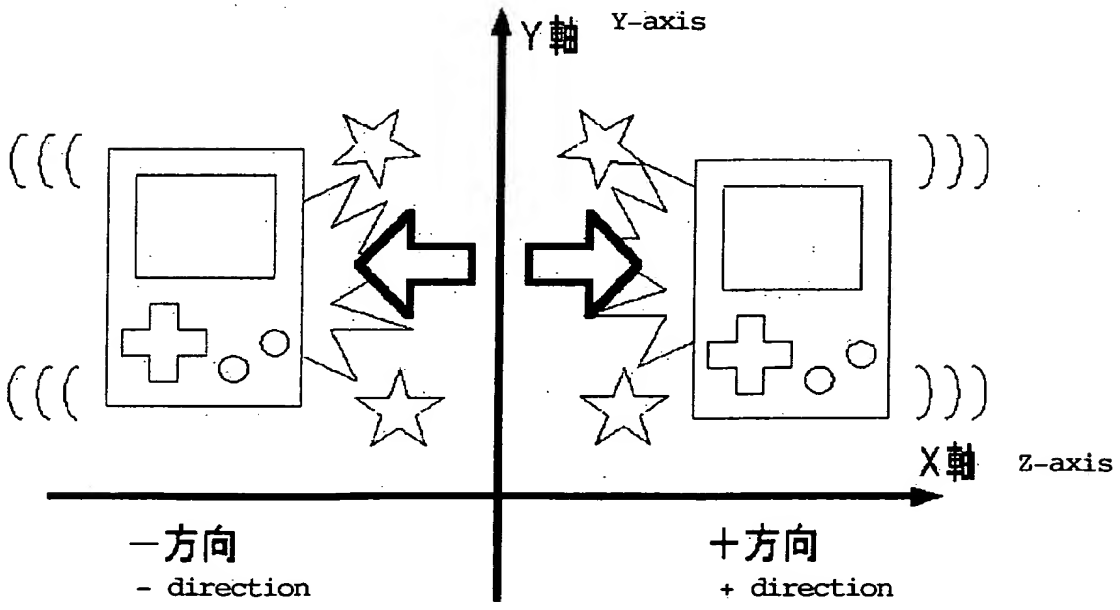
【図１１】 [Figure 11]



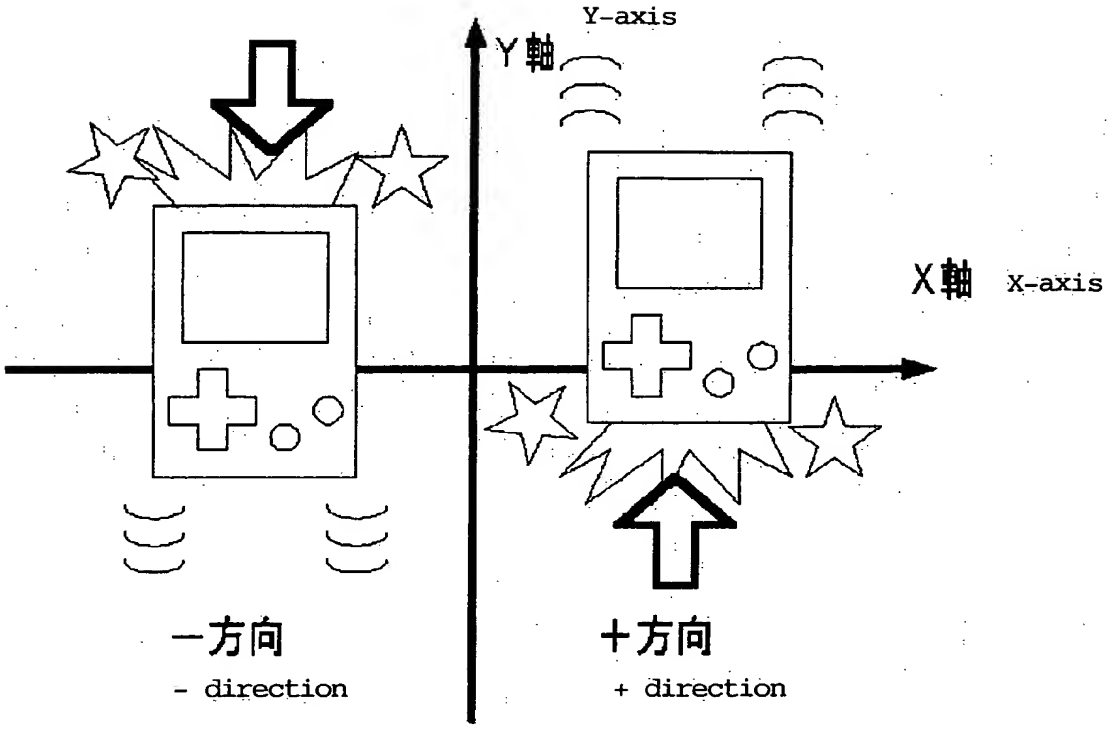
【図１２】 [Figure 12]



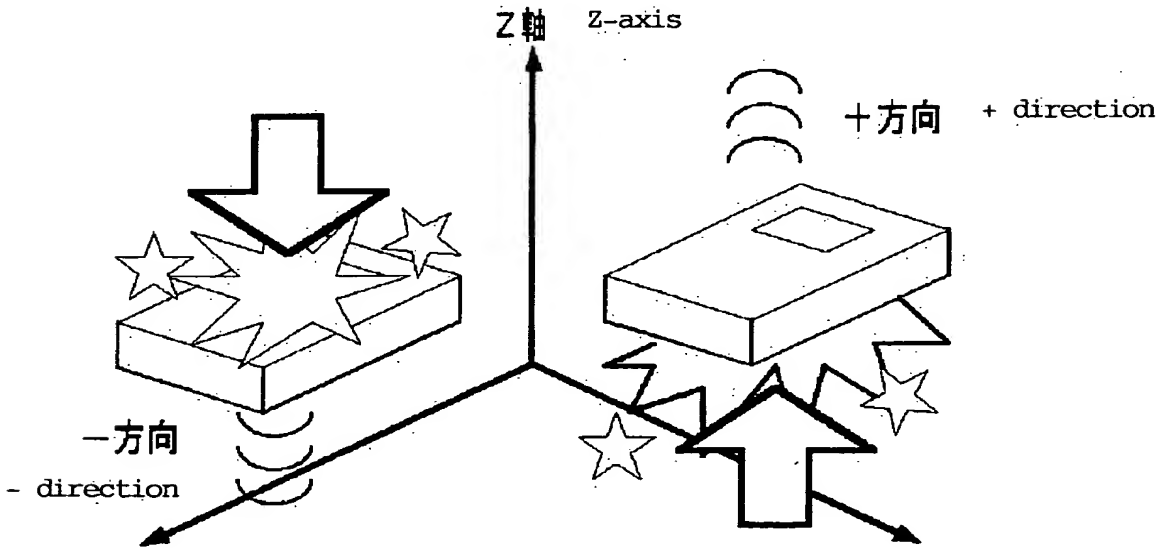
【図１３】 [Figure 13]



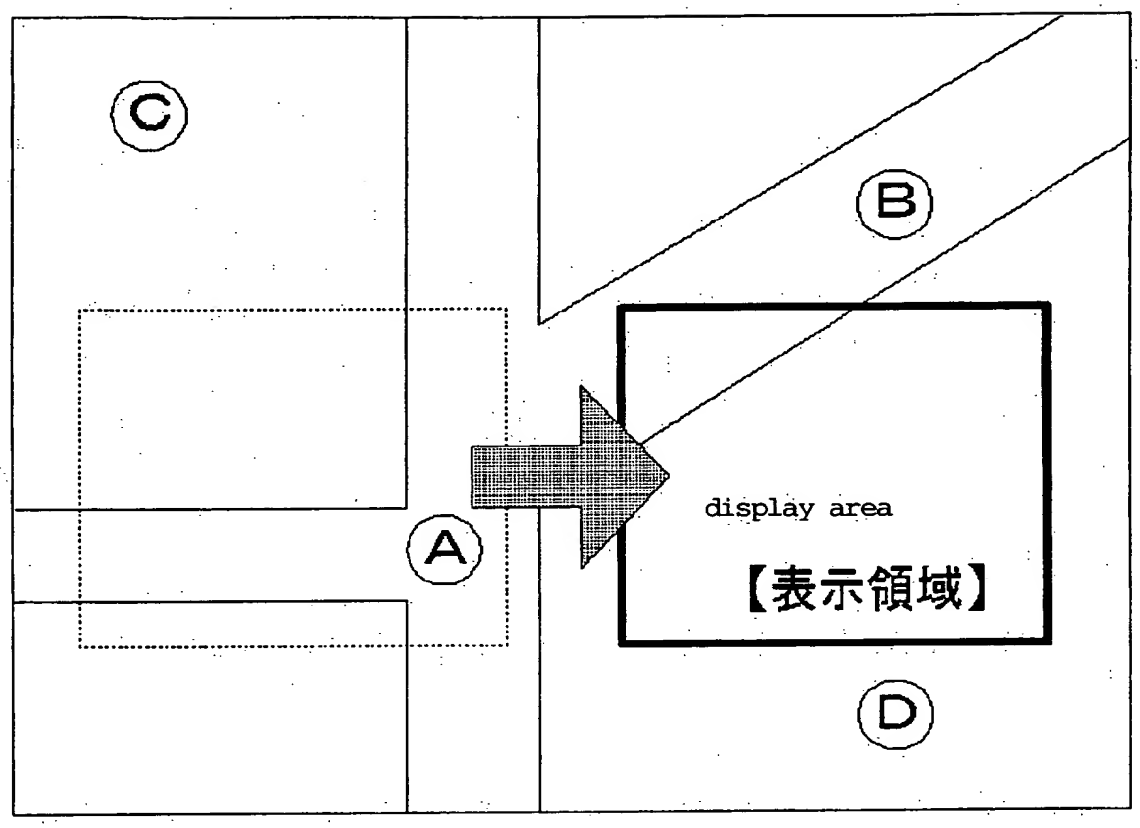
【図１４】 [Figure 14]



【図１５】 [Figure 15]

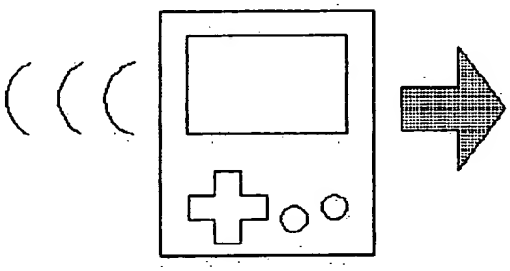


【図１６】 [Figure 16]



【仮想マップ】

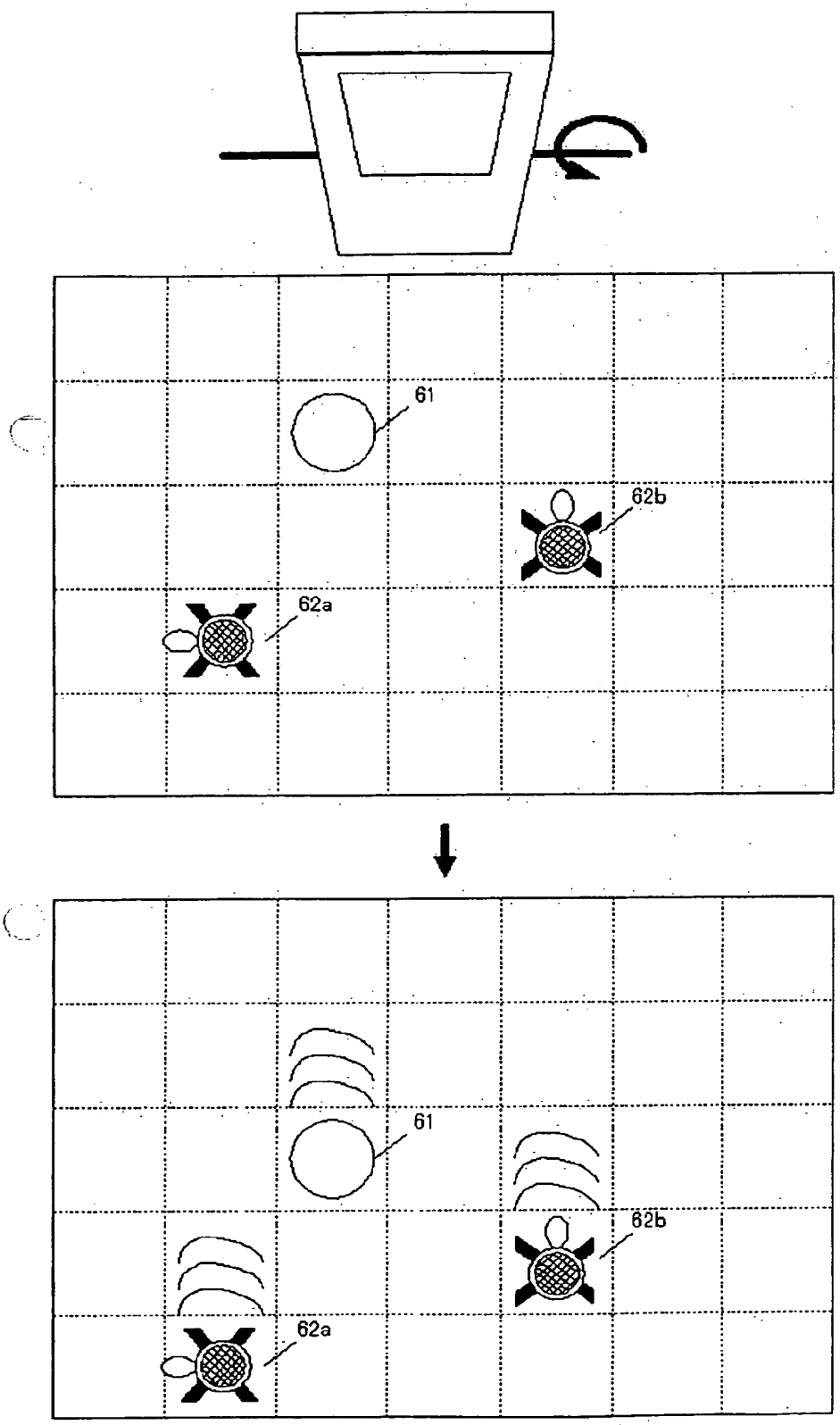
virtual map



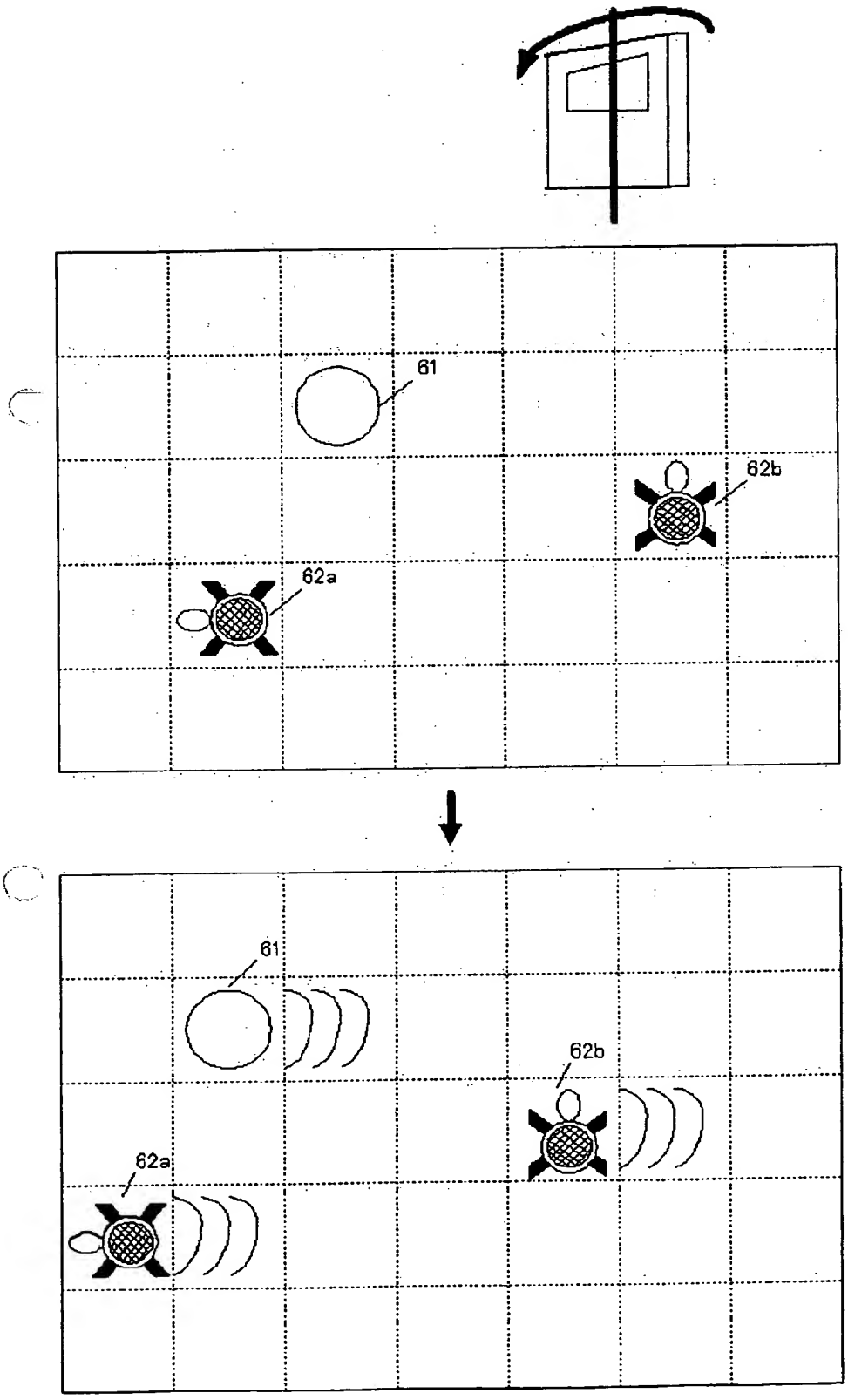
【スライド】

slide

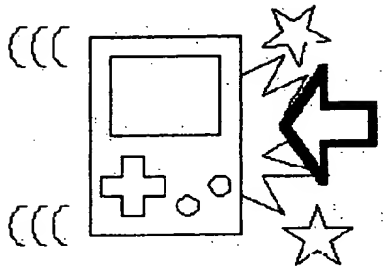
【図１７】 [Figure 17]



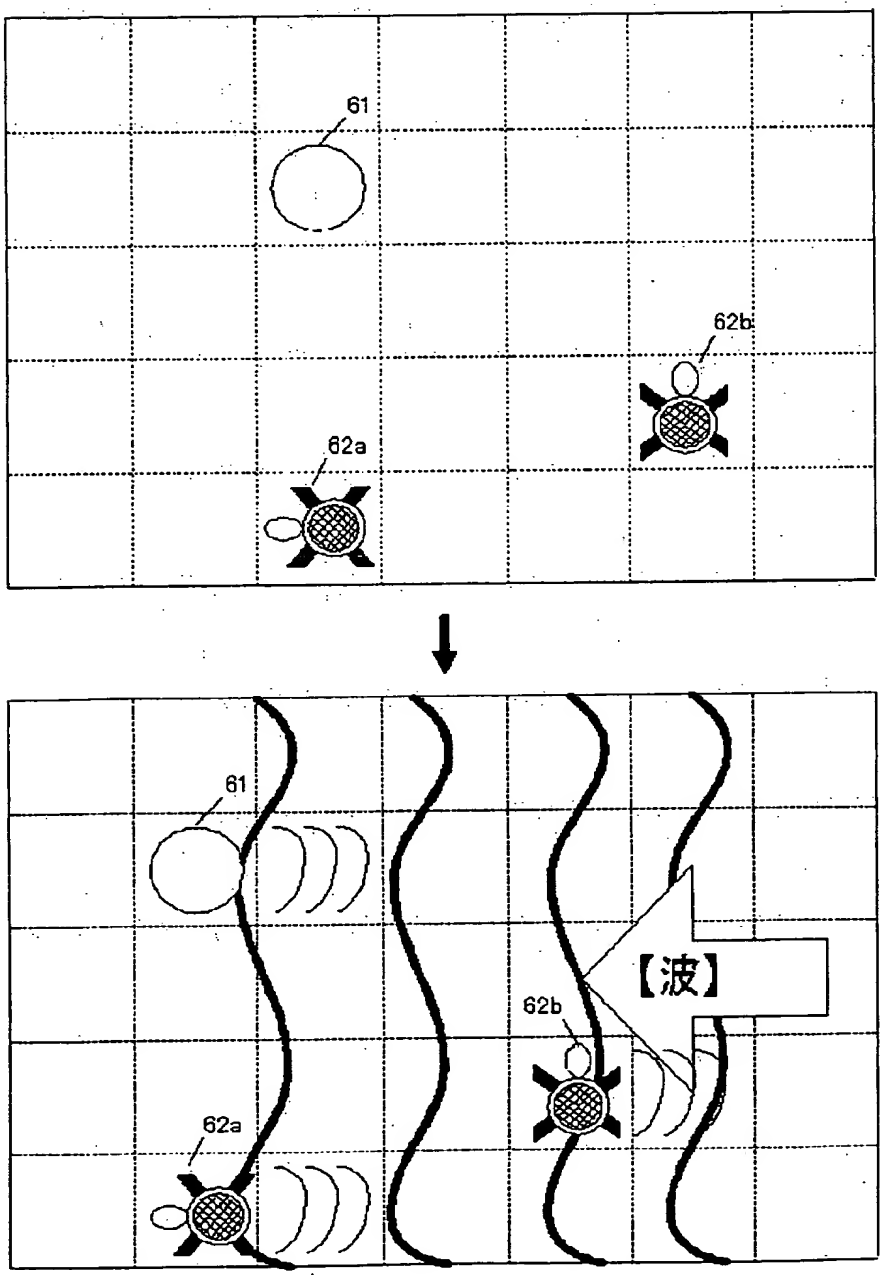
【図１８】 [Figure 18]



【図１９】 [Figure 19]

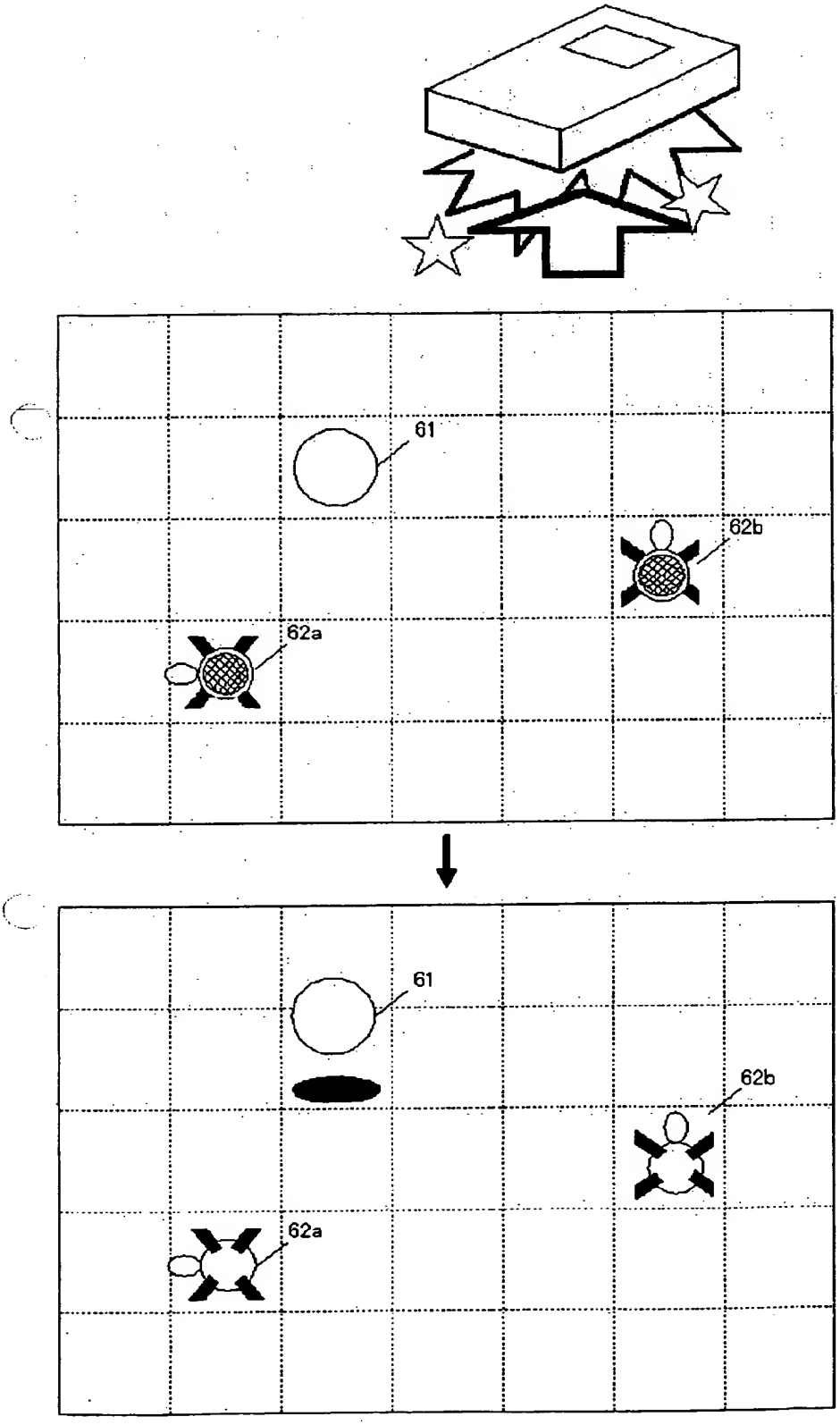


衝撃入力 impact input

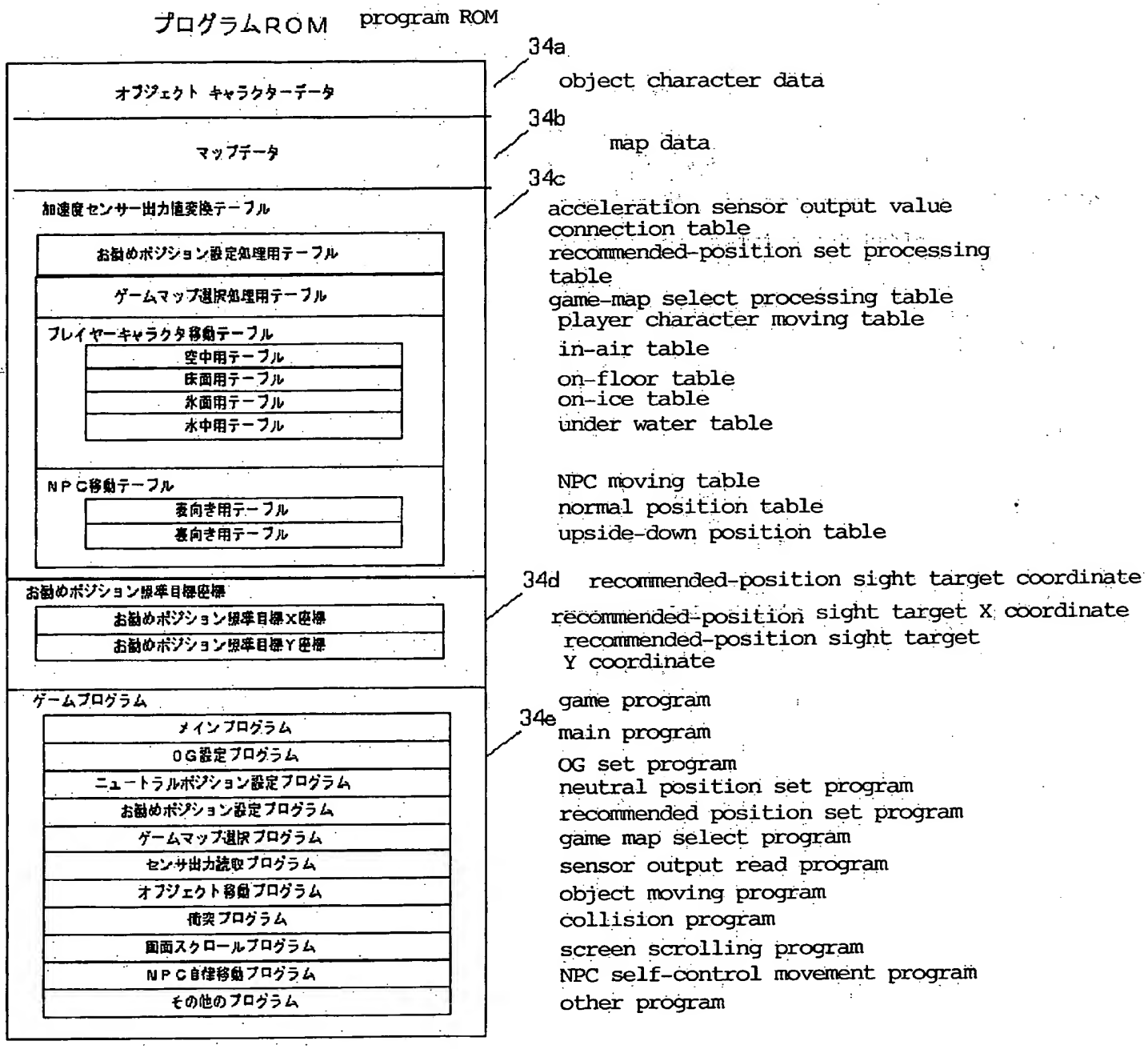


wave

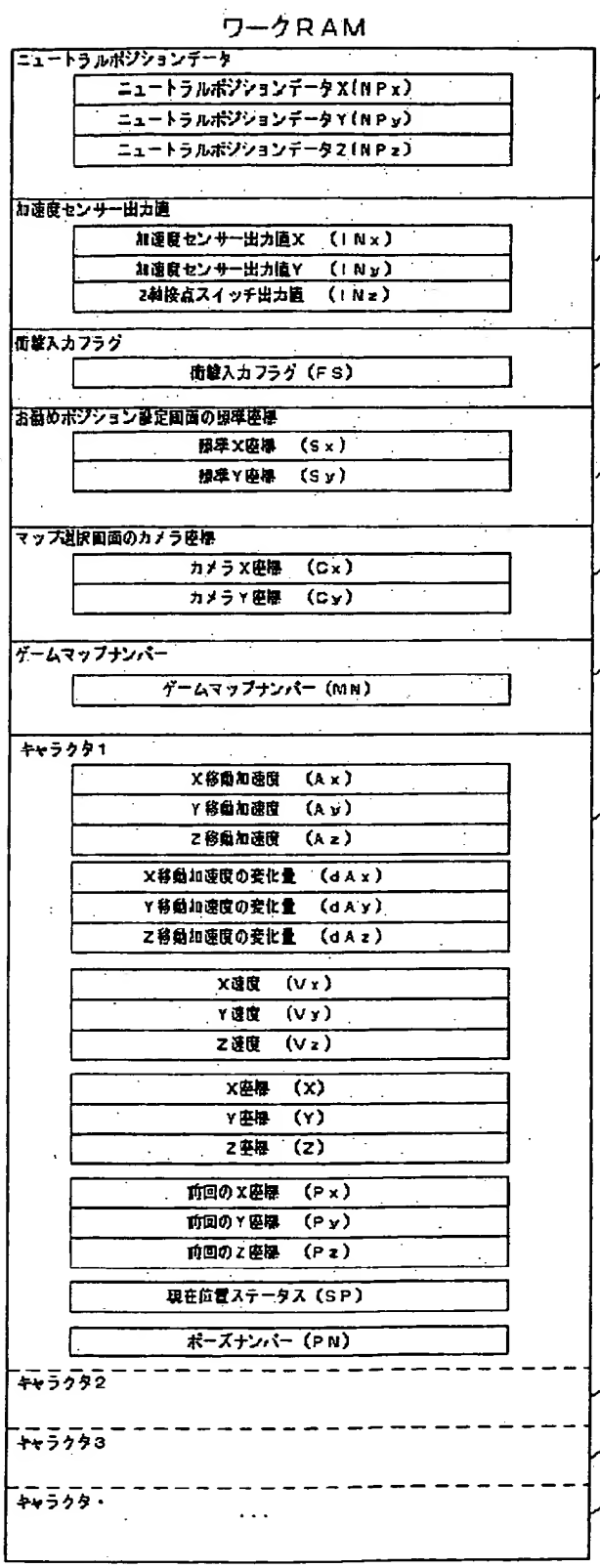
【図２０】 [Figure 20]



【図２１】 [Figure 21]



【図２２】 [Figure 22]



work RAM

neutral position data
neutral position data X
neutral position data Y
neutral position data Z

acceleration sensor output values
acceleration sensor output value X
acceleration sensor output value Y
Z-axis contact switch output value

impact input flag
impact input flag

sight coordinate on recommended-position set scene
sight X coordinate
sight Y coordinate

camera coordinates of map select scene
camera X coordinate
camera Y coordinate

game map number
game map number

character 1
X moving acceleration
Y moving acceleration
Z moving acceleration

X moving acceleration change amount
Y moving acceleration change amount
Z moving acceleration change amount

X velocity
Y velocity
Z velocity

X coordinate
Y coordinate
Z coordinate

last-time X coordinate
last-time Y coordinate
last-time Z coordinate

current position status

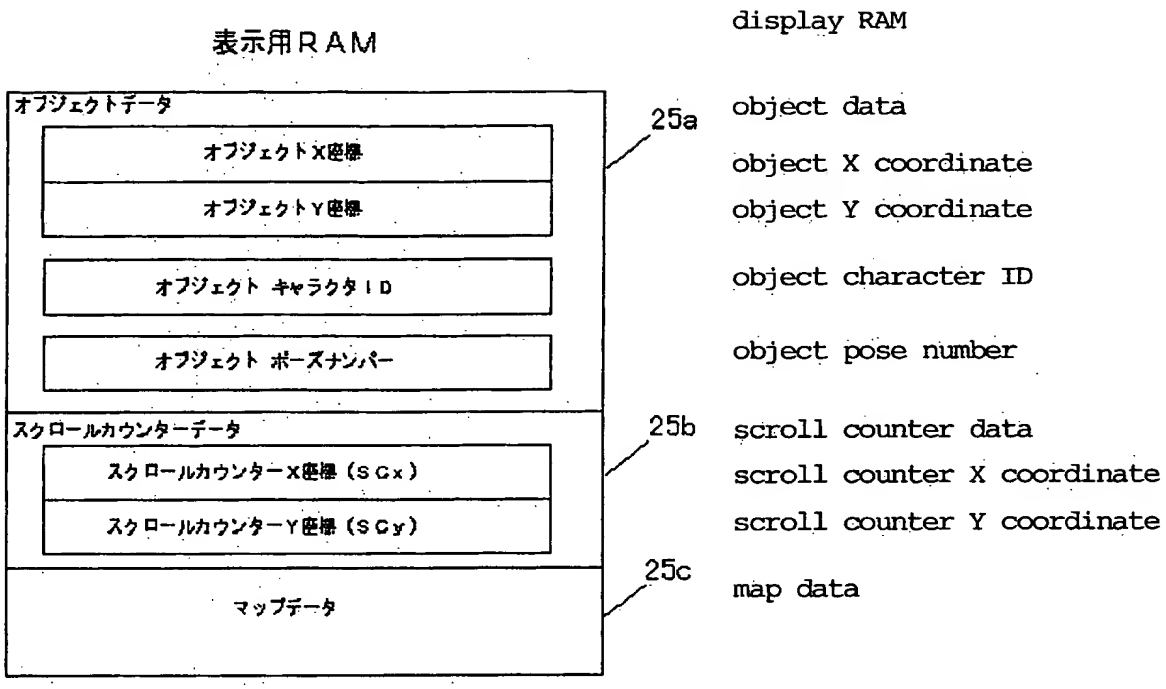
pose number

character 2

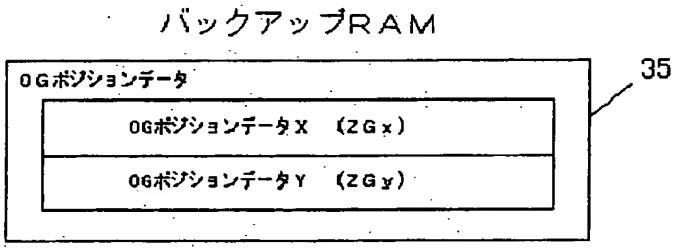
character 3

character

【図２３】 [Figure 23]



【図２４】 [Figure 24]



【図２５】 [Figure 25]

お勧めポジション設定処理用テーブル

utilization method

利用方法	補正比率	特殊補正条件1	特殊補正数1	特殊補正条件2	特殊補正数2
センサ出力値X（INx）	照準X座標（Sx）	×1	なし	なし	なし	なし
センサ出力値Y（INy）	照準Y座標（Sy）	×1	なし	なし	なし	なし
Z軸接点スイッチ出力値（INz）	無視 ignored	なし	なし	なし	なし	なし
衝撃入力フラグ（FS）	無視 ignored	なし	なし	なし	なし	なし

sight X coordinate (Sx)
sight Y coordinate (Sy)

sensor output value X
sensor output value Y
Z-axis contact switch output value
impact input flag

correction ratio
particular correction number 1
particular correction number 2
particular correction condition 1
particular correction condition 2

【図２６】

ゲームマップ選択処理用テーブル

	利用方法	補正比率	特殊補正条件1	特殊補正数1	特殊補正条件2	特殊補正数2
センサ出力値X (INx)	カメラX座標(Cx)の変化量	x2	なし	なし	なし	なし
センサ出力値Y (INy)	カメラY座標(Cy)の変化量	x2	なし	なし	なし	なし
Z軸接点スイッチ出力値(INz)	マップ決定	なし	なし	なし	なし	なし
衝撃入力フラグ (FS)	無視	なし	なし	なし	なし	なし

【図２７】

プレイヤーキャラクタ移動テーブル（空中用）

	利用方法	補正比率	特殊補正条件1	特殊補正数1	特殊補正条件2	特殊補正数2
センサ出力値X (INx)	無視	なし	なし	なし	なし	なし
センサ出力値Y (INy)	無視	なし	なし	なし	なし	なし
Z軸接点スイッチ出力値(INz)	Z移動加速度の変化量(dAz)	x1	なし	なし	なし	なし
衝撃入力フラグ (FS)	無視	なし	なし	なし	なし	なし

【図２８】

プレイヤーキャラクタ移動テーブル（床面用）

	利用方法	補正比率	特殊補正条件1	特殊補正数1	特殊補正条件2	特殊補正数2
センサ出力値X (INx)	X移動加速度の変化量(dAx)	x2	INx>20	40	なし	なし
センサ出力値Y (INy)	Y移動加速度の変化量(dAy)	x2	INy>20	40	なし	なし
Z軸接点スイッチ出力値(INz)	Z移動加速度の変化量(dAz)	x1	なし	なし	なし	なし
衝撃入力フラグ (FS)	X,Y移動加速度の変化量(dAx, dAy)	x3	なし	なし	なし	なし

【図２９】

プレイヤーキャラクタ移動テーブル（氷面用）

	利用方法	補正比率	特殊補正条件1	特殊補正数1	特殊補正条件2	特殊補正数2
センサ出力値X (INx)	X移動加速度の変化量(dAx)	x3	INx>20	60	なし	なし
センサ出力値Y (INy)	Y移動加速度の変化量(dAy)	x3	INy>20	60	なし	なし
Z軸接点スイッチ出力値(INz)	Z移動加速度の変化量(dAz)	x1	なし	なし	なし	なし
衝撃入力フラグ (FS)	X,Y移動加速度の変化量(dAx, dAy)	x3	なし	なし	なし	なし

【図２６】 [Figure 26]

game map select processing table

	utilization method	correction ratio	particular correction condition 1	particular correction number 1	particular correction condition 2	particular correction number 2
sensor output value X (INx)	change amount of X moving acceleration (Cx)	×2	–	–	–	–
sensor output value Y (INy)	change amount of Y moving acceleration (Cy)	×2	–	–	–	–
Z-axis contact SW output vlue (INz)	map decision	–	–	–	–	–
impact input flag (FS)	–	–	–	–	–	–

【図２７】 [Figure 27]

player character moving table (in-air)

	utilization method	correction ratio	particular correction condition 1	particular correction number 1	particular correction condition 2	particular correction number 2
sensor output value X (INx)	–	–	–	–	–	–
sensor output value Y (INy)	–	–	–	–	–	–
Z-axis contact SW output vlue (INz)	change amount of Z moving acceleration (dAz)	×1	–	–	–	–
impact input flag (FS)	–	–	–	–	–	–

【図２８】 [Figure 28]

player character moving table (on-ice)

	utilization method	correction ratio	particular correction condition 1	particular correction number 1	particular correction condition 2	particular correction number 2
sensor output value X (INx)	change amount of X moving acceleration (dAx)	×3	INx>20	60	–	–
sensor output value Y (INy)	change amount of Y moving acceleration (dAy)	×3	INy>20	60	–	–
Z-axis contact SW output vlue (INz)	change amount of Z moving acceleration (dAz)	×1	–	–	–	–
impact input flag (FS)	change amount of XY moving acceleration (dAx, dAy)	×5	–	–	–	–

【図２９】 [Figure 29]

player character moving table (on-floor)

	utilization method	correction ratio	particular correction condition 1	particular correction number 1	particular correction condition 2	particular correction number 2
sensor output value X (INx)	change amount of X moving acceleration (dAx)	×2	INx>20	40	–	–
sensor output value Y (INy)	change amount of Y moving acceleration (dAy)	×2	INy>20	40	–	–
Z-axis contact SW output vlue (INz)	change amount of Z moving acceleration (dAz)	×1	–	–	–	–
impact input flag (FS)	change amount of XY moving acceleration (dAx, dAy)	×3	–	–	–	–

【図３０】

プレイヤーキャラクタ移動テーブル（水中用）

	利用方法	補正比率	特殊補正条件1	特殊補正量1	特殊補正条件2	特殊補正量2
センサ出力値X (INx)	X移動加速度の変化量(dAx)	× 1/2	INx＞10	5	なし	なし
センサ出力値Y (INy)	Y移動加速度の変化量(dAy)	× 1/2	INy＞10	5	なし	なし
Z軸接点スイッチ出力値(INz)	Z移動加速度の変化量(dAz)	x1	なし	なし	なし	なし
衝撃入力フラグ (FS)	無視	なし	なし	なし	なし	なし

【図３１】

ＮＰＣ移動テーブル（亀表向き用）

	利用方法	補正比率	特殊補正条件1	特殊補正量1	特殊補正条件2	特殊補正量2
センサ出力値X (INx)	X移動加速度の変化量(dAx)	× 1/2	INx＜10	0	INx＞20	10
センサ出力値Y (INy)	Y移動加速度の変化量(dAy)	× 1/2	INy＜10	0	INy＞20	10
Z軸接点スイッチ出力値(INz)	表裏逆転	なし	なし	なし	なし	なし
衝撃入力フラグ (FS)	無視	なし	なし	なし	なし	なし

【図３２】

ＮＰＣ移動テーブル（亀裏向き用）

	利用方法	補正比率	特殊補正条件1	特殊補正量1	特殊補正条件2	特殊補正量2
センサ出力値X (INx)	X移動加速度の変化量(dAx)	x2	INx＞20	40	なし	なし
センサ出力値Y (INy)	Y移動加速度の変化量(dAy)	x2	INy＞20	40	なし	なし
Z軸接点スイッチ出力値(INz)	表裏逆転	なし	なし	なし	なし	なし
衝撃入力フラグ (FS)	無視	なし	なし	なし	なし	なし

【図３０】 [Figure 30]

player character moving table (under-water)

sensor output value X (INx)	utilization method	correction ratio	particular correction condition 1	particular correction number 1	particular correction condition 2	particular correction number 2
sensor output value Y (INy)	change amount of X moving acceleration (dAx)	×1/2	INx>10	5	–	–
Z-axis contact SW output vlue (INz)	change amount of Y moving acceleration (dAy)	×1/2	INy>10	5	–	–
impact input flag (FS)	change amount of Z moving acceleration (dAz)	×1	–	–	–	–
	–	–	–	–	–	–

【図３１】 [Figure 31]

NPC moving table (for tortoise normal position)

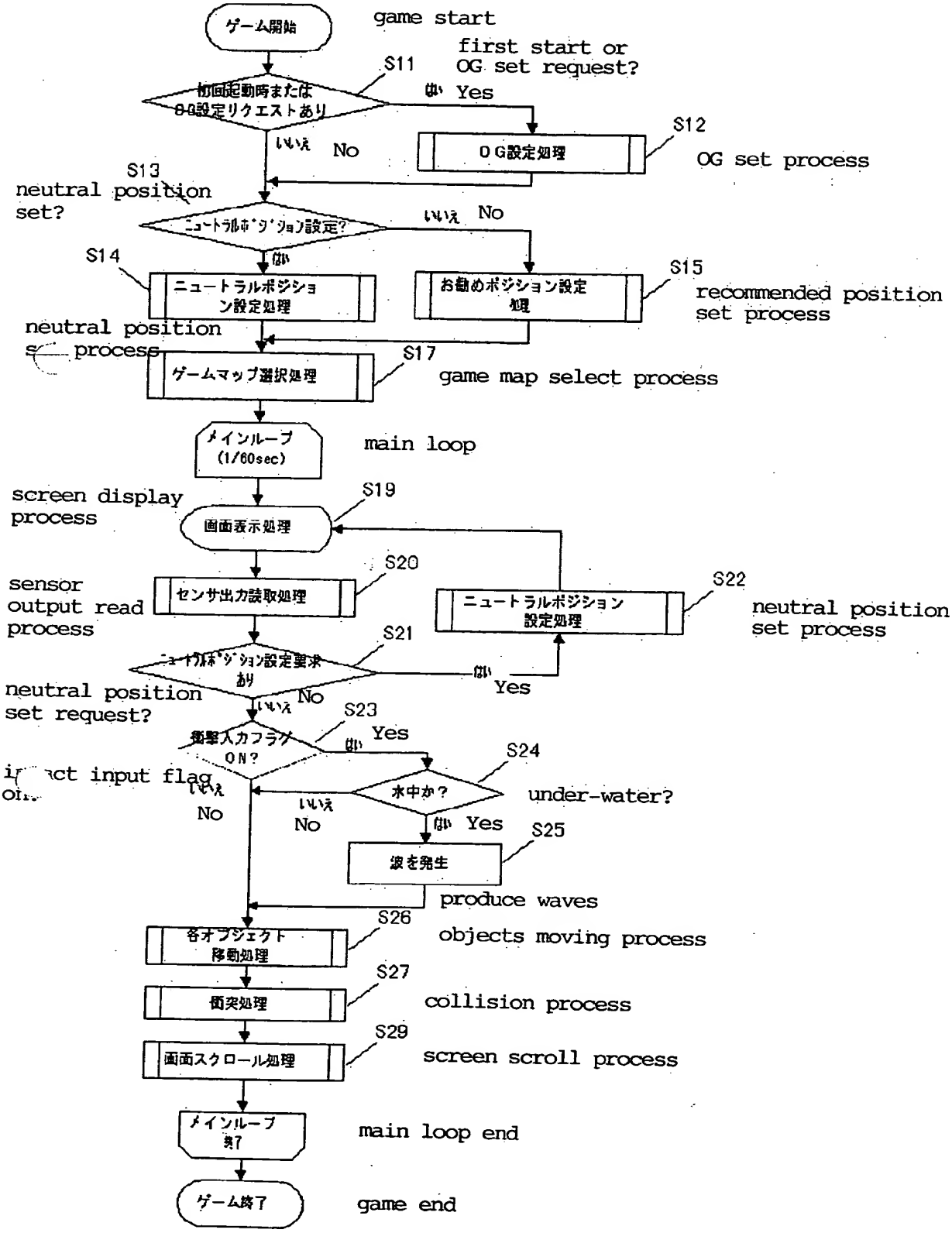
sensor output value X (INx)	utilization method	correction ratio	particular correction condition 1	particular correction number 1	particular correction condition 2	particular correction number 2
sensor output value Y (INy)	change amount of X moving acceleration (dAx)	×1/2	INx<10	0	INx>20	10
Z-axis contact SW output vlue (INz)	change amount of Y moving acceleration (dAy)	×1/2	INy<10	0	INy>20	10
impact input flag (FS)	position inversion	–	–	–	–	–
	–	–	–	–	–	–

【図３２】 [Figure 32]

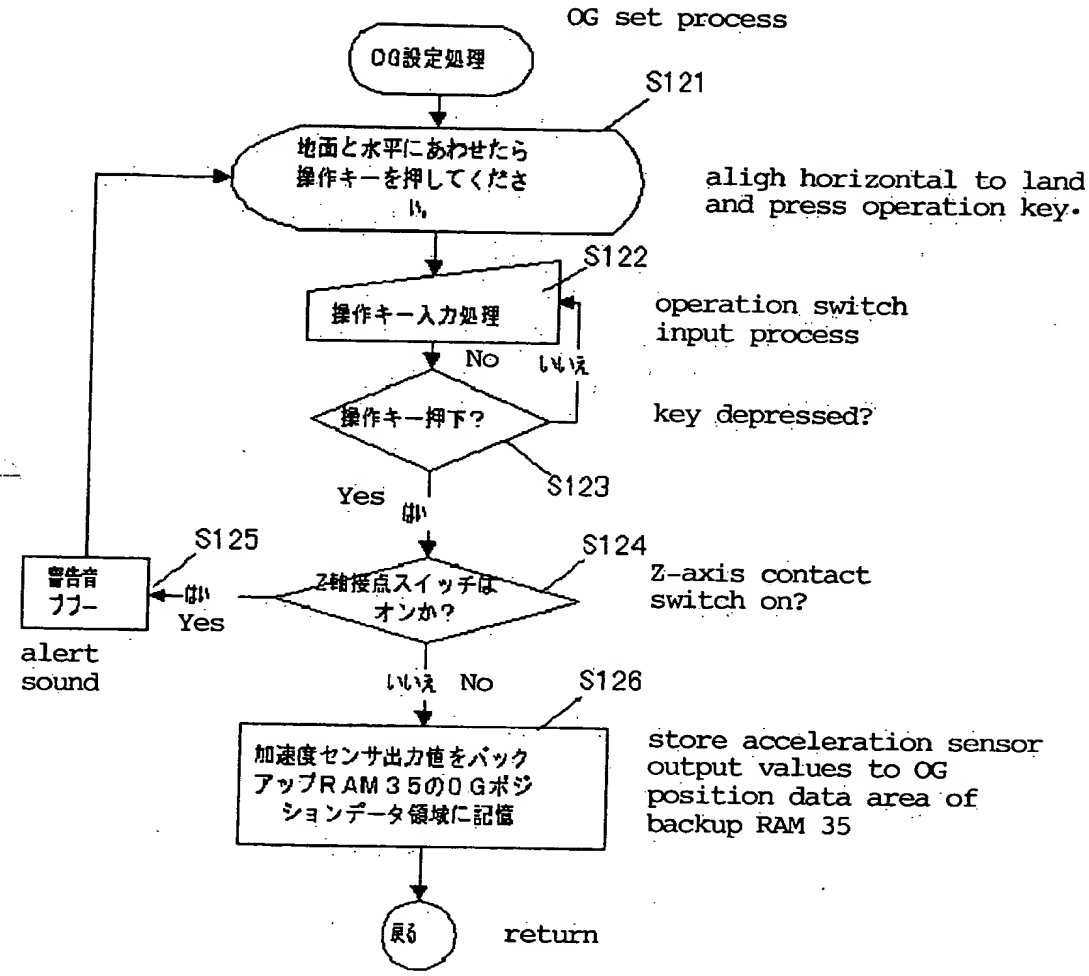
NPC moving table (for tortoise upside-down position)

sensor output value X (INx)	utilization method	correction ratio	particular correction condition 1	particular correction number 1	particular correction condition 2	particular correction number 2
sensor output value Y (INy)	change amount of X moving acceleration (dAx)	×2	INx>20	40	–	–
Z-axis contact SW output vlue (INz)	change amount of Y moving acceleration (dAy)	×2	INy>20	40	–	–
impact input flag (FS)	position inversion	–	–	–	–	–
	–	–	–	–	–	–

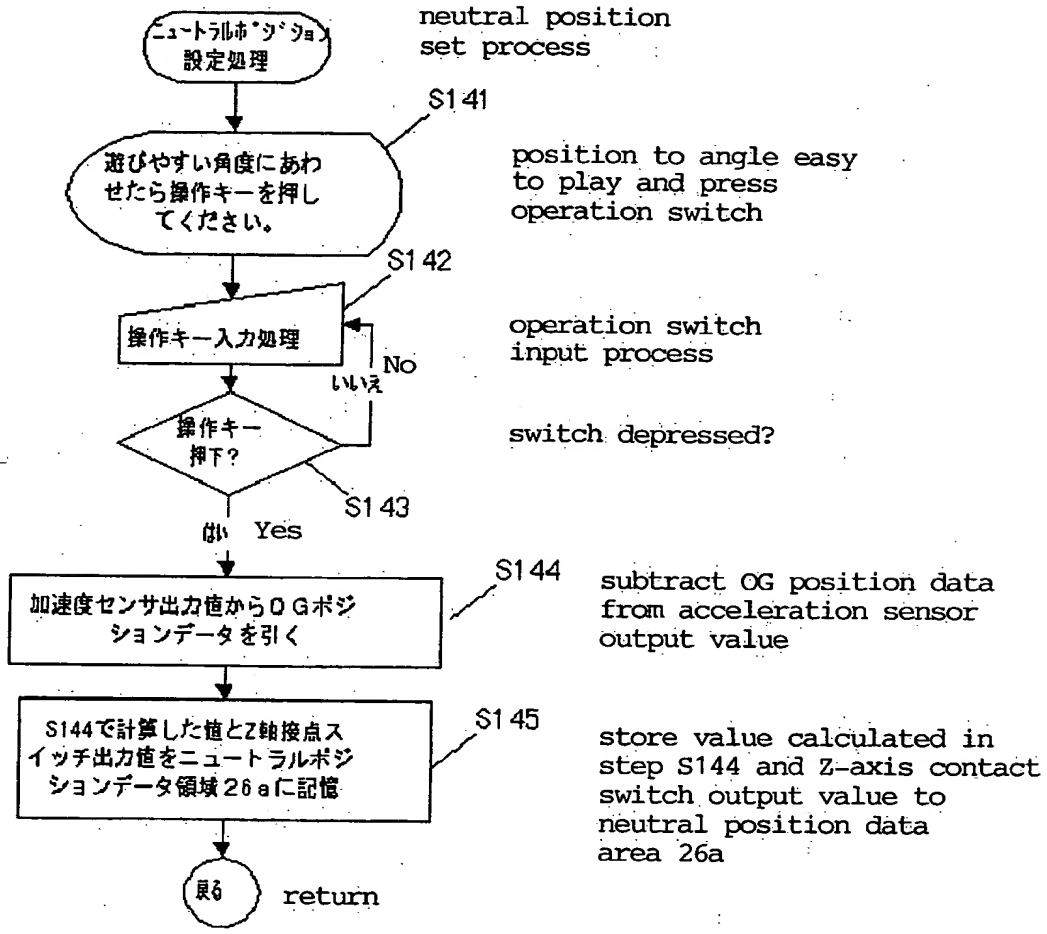
【図３３】 [Figure 33]



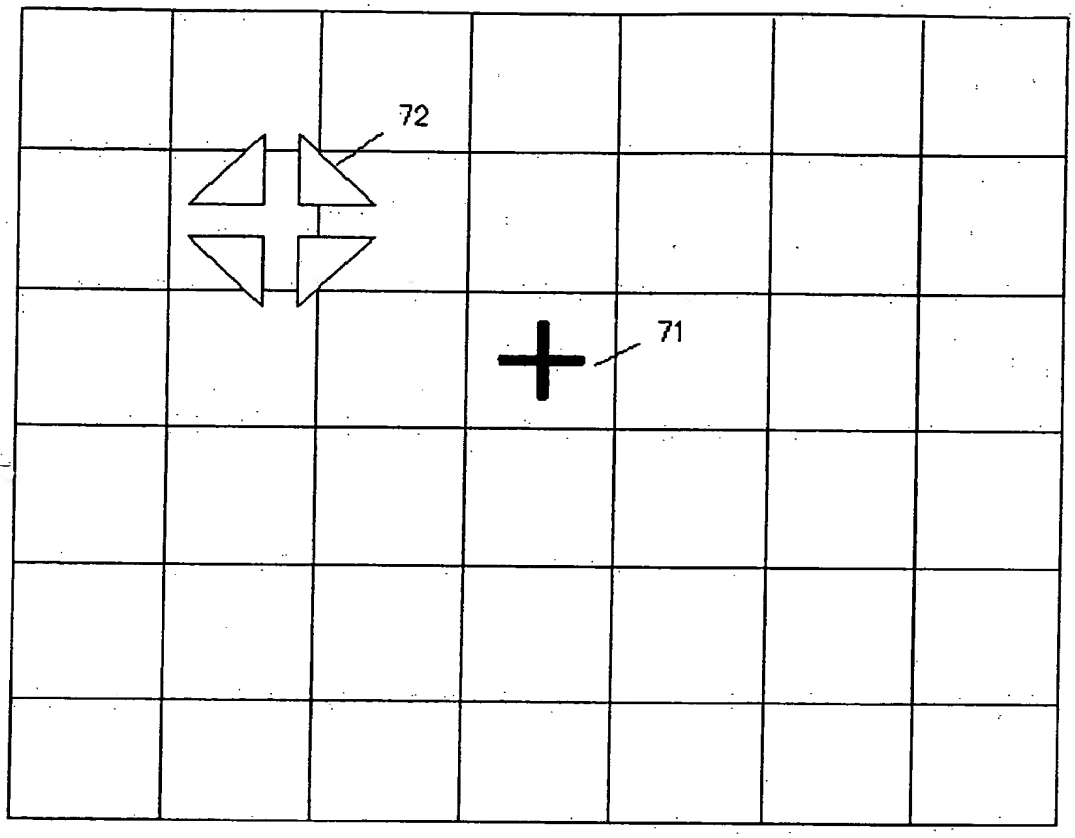
【図３４】 [Figure 34]



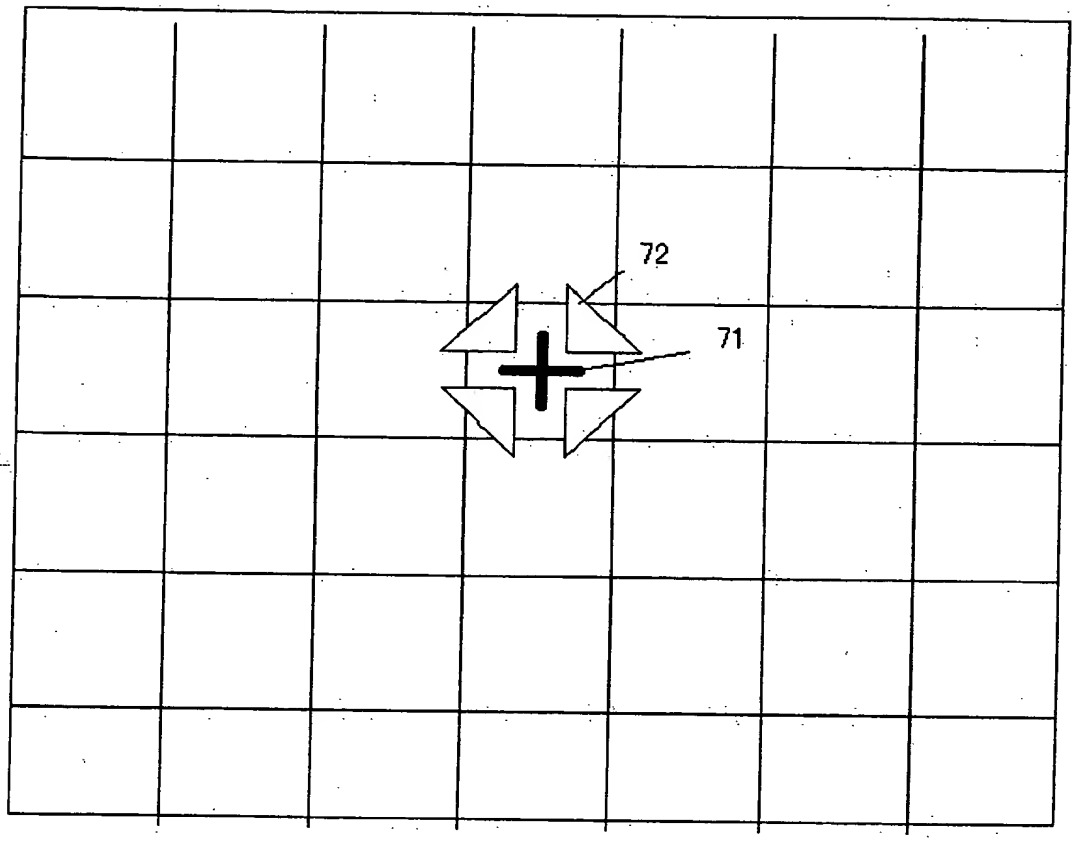
【図３５】 [Figure 35]



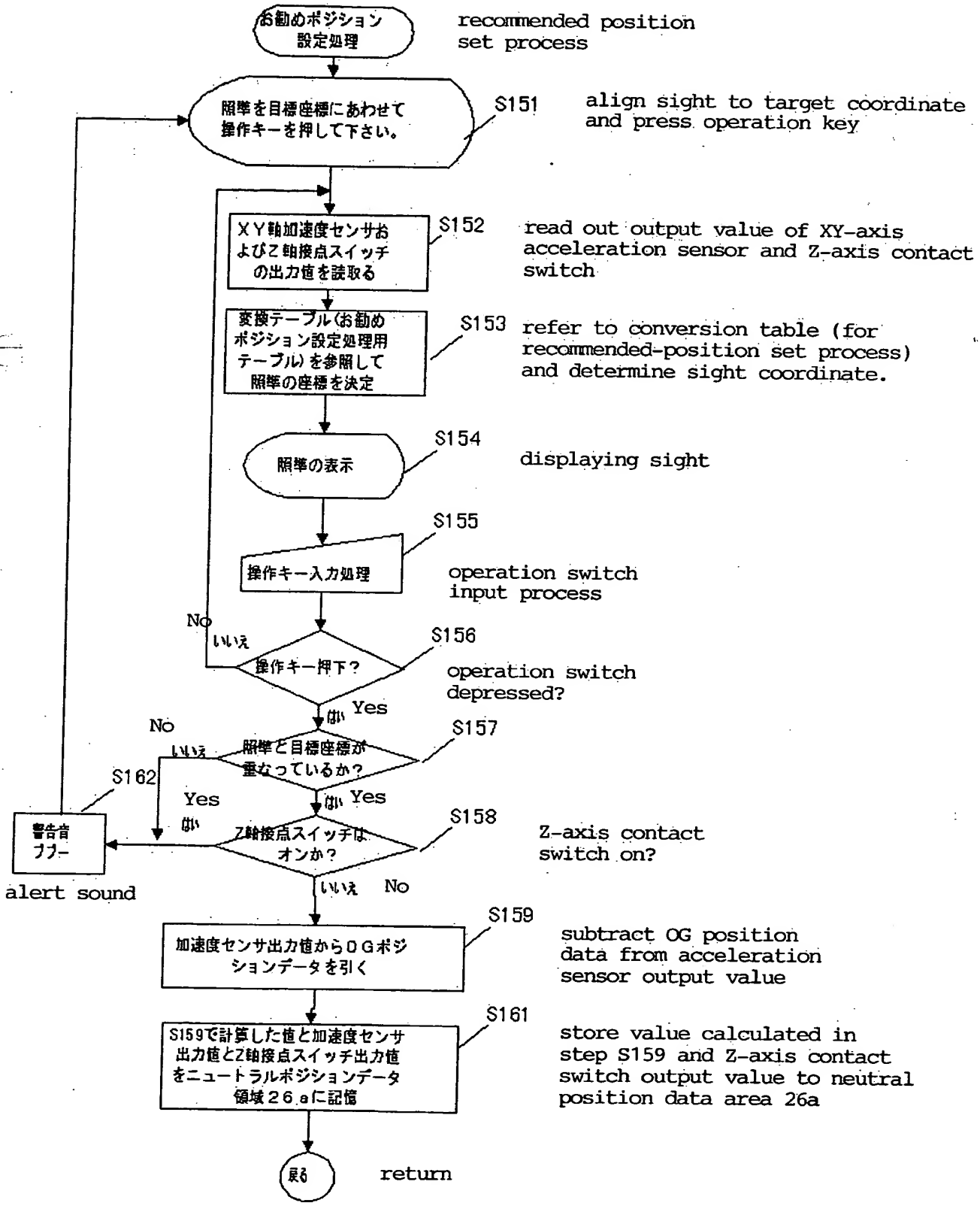
【図３６】 [Figure 36]



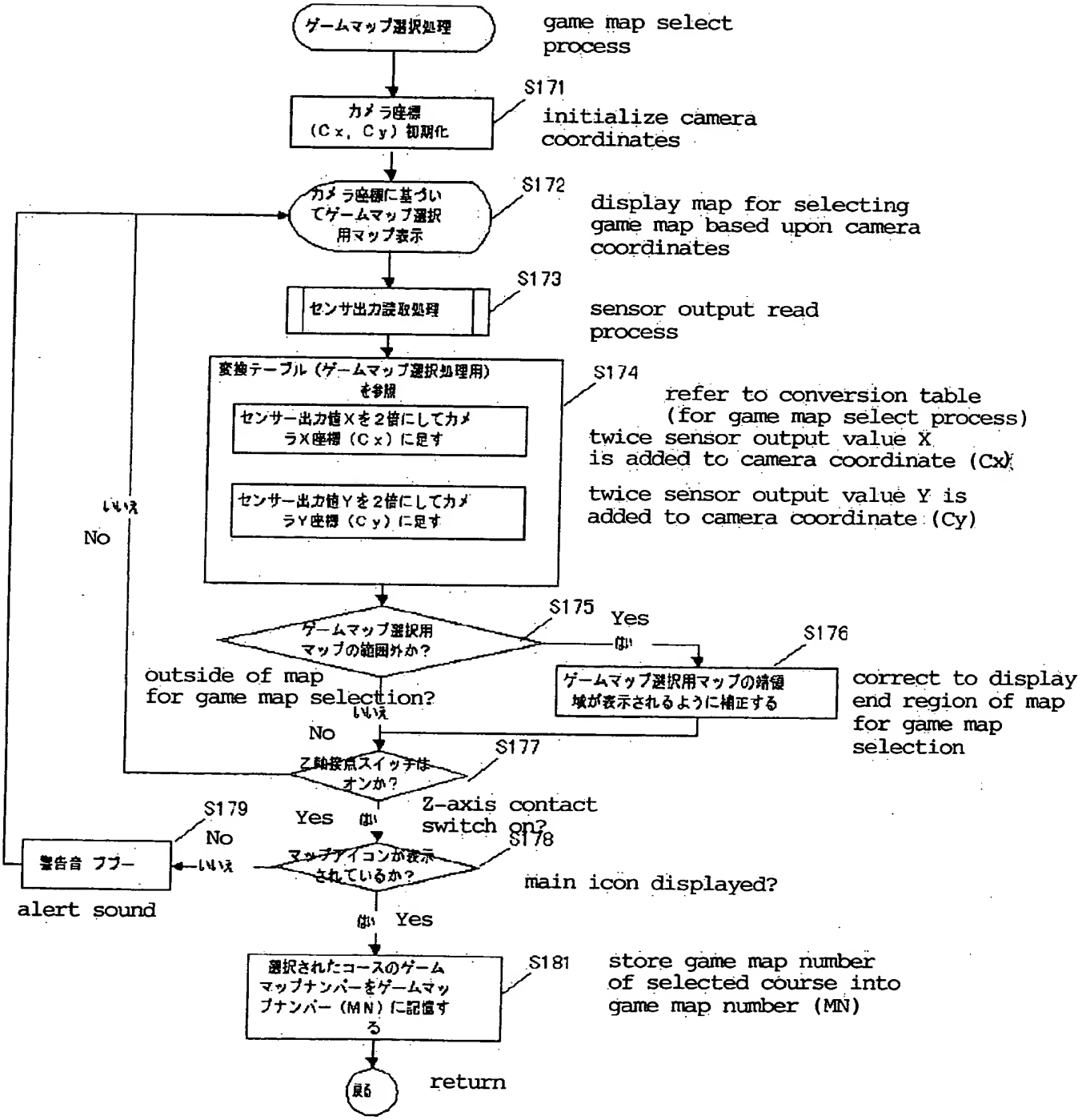
【図３７】　[Figure 37]



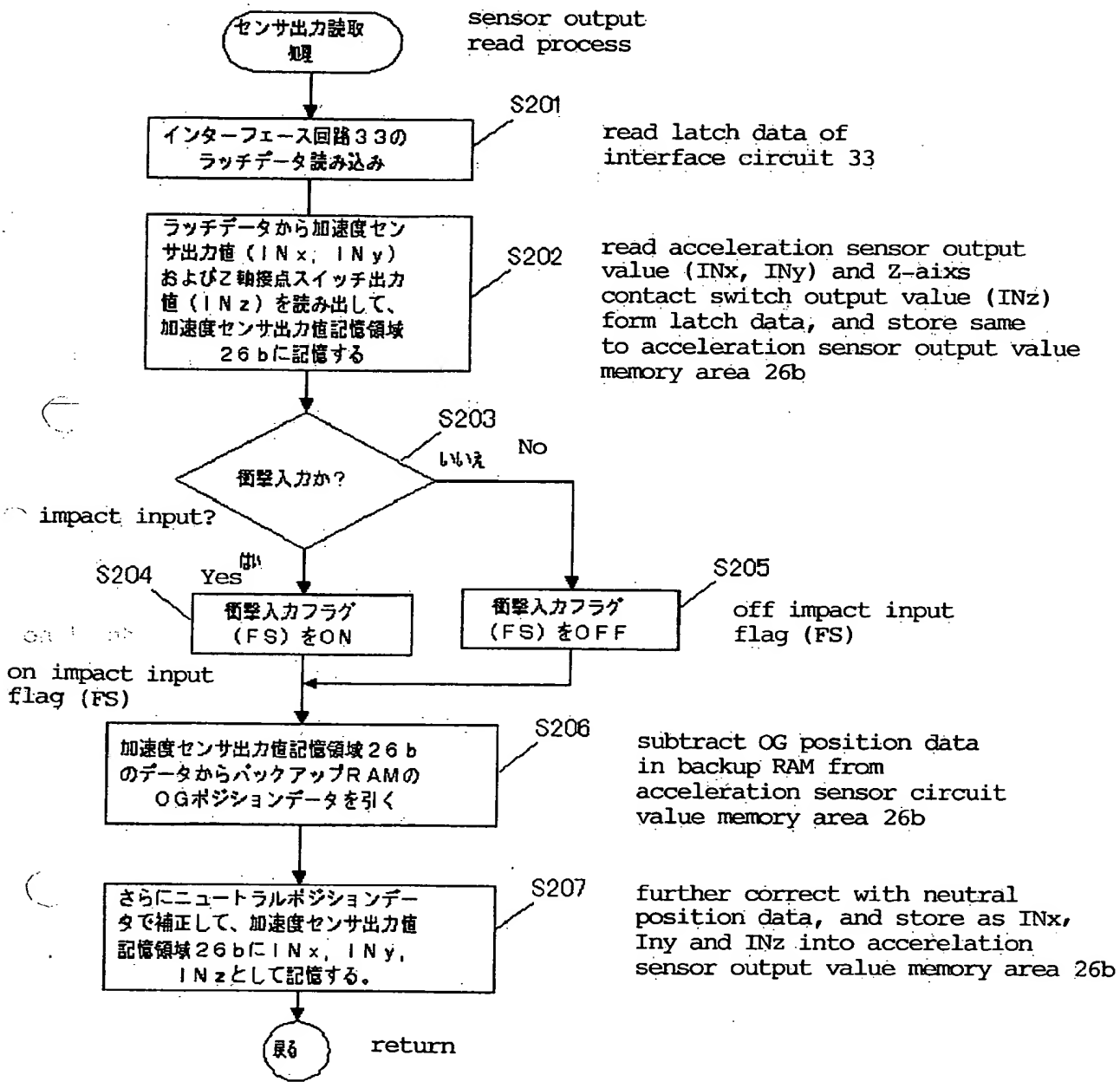
【図３８】 [Figure 38]



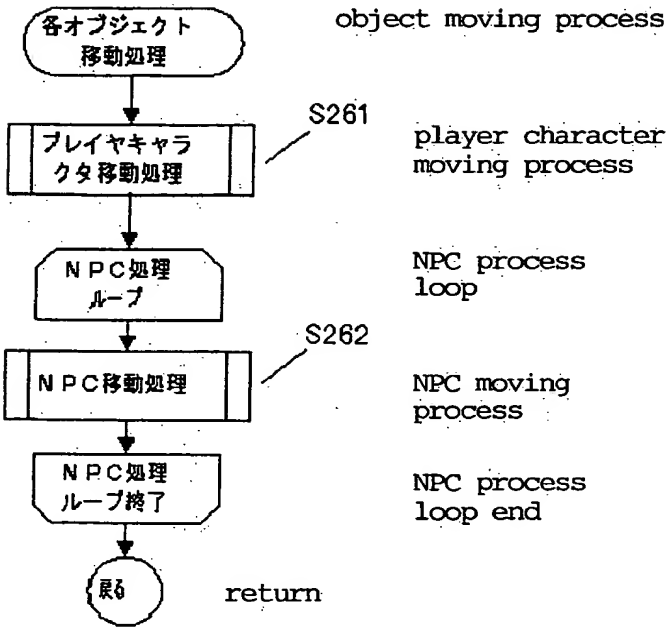
【図３９】 [Figure 39]



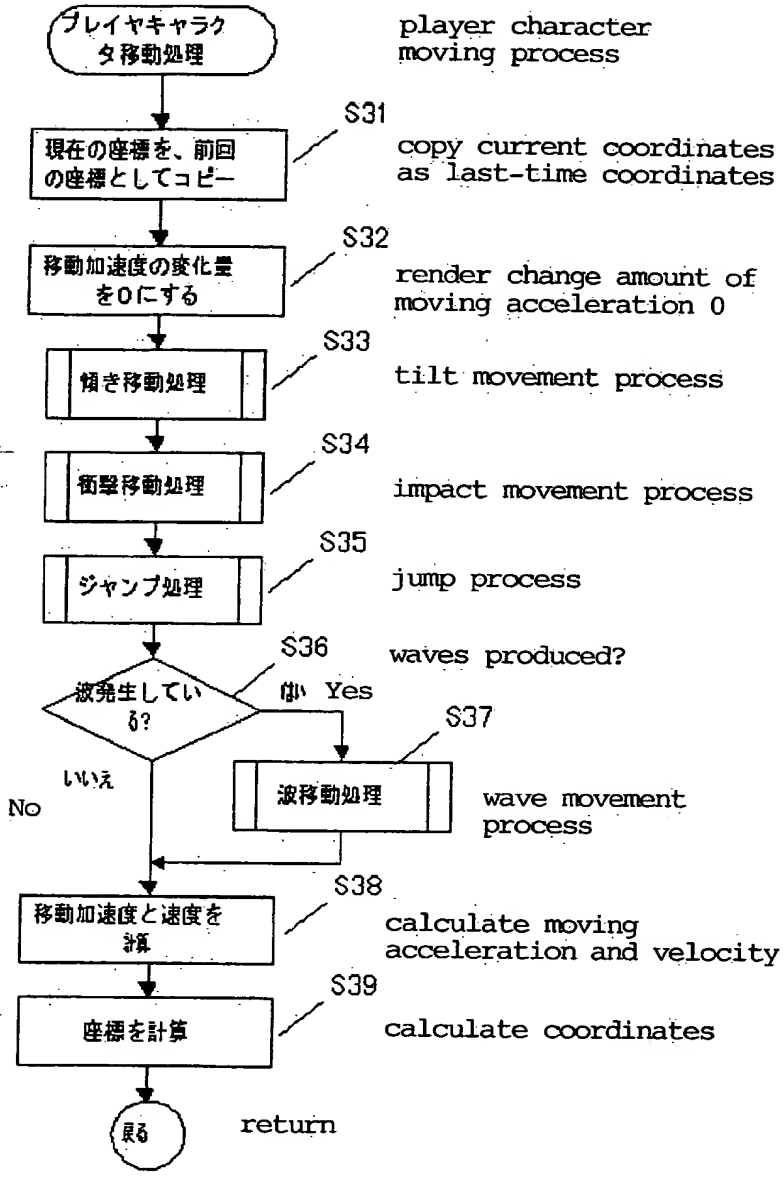
【図４０】 [Figure 40]



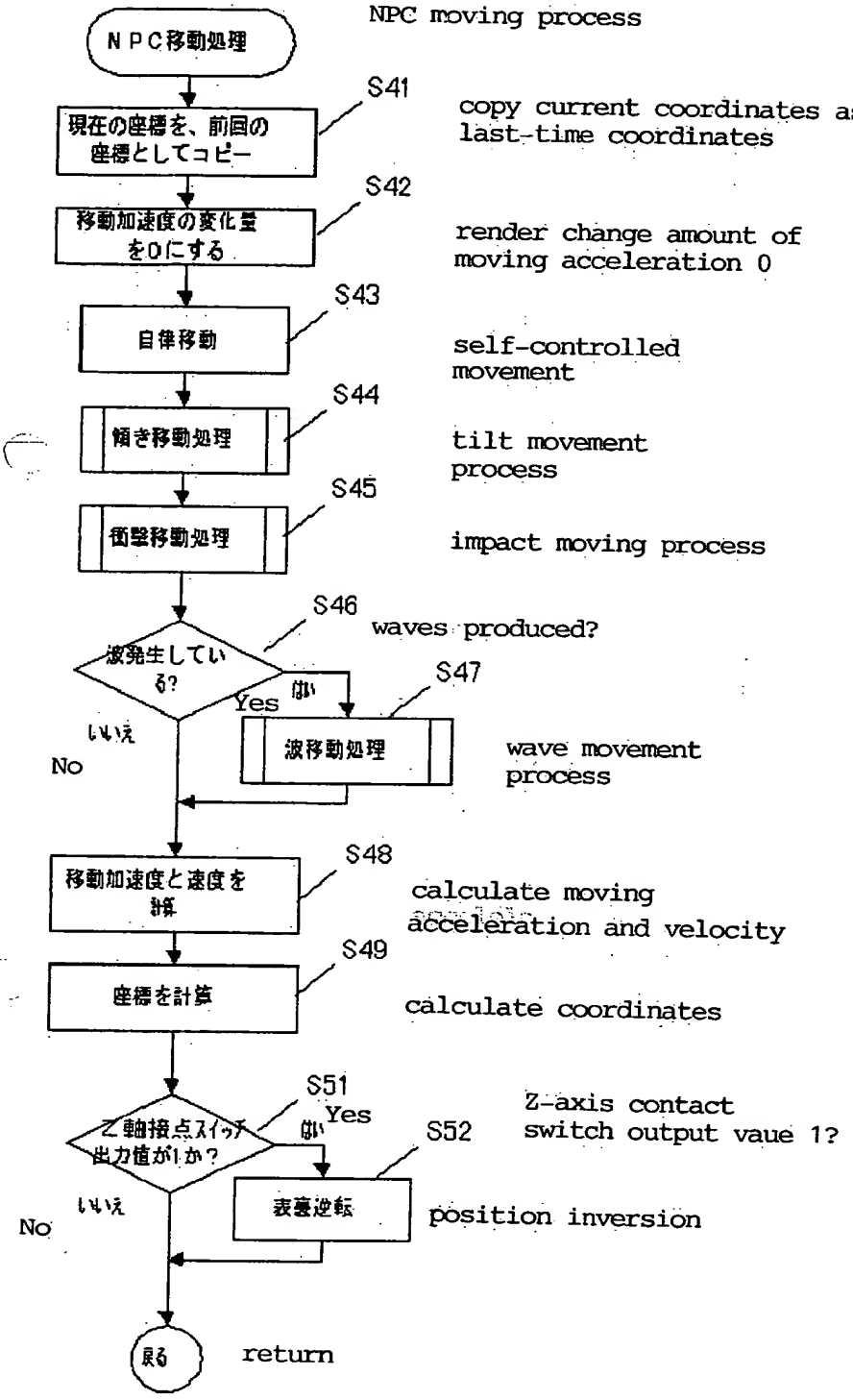
【図４１】 [Figure 41]



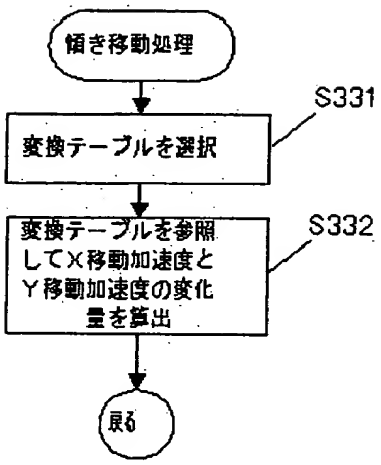
【図４２】 [Figure 42]



【図４３】 [Figure 43]



【図４４】



[Figure 44]

tilt movement process

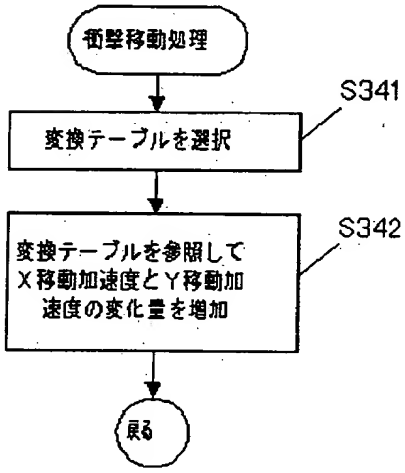
select conversion table

refer to conversion table and calculate change of X movement acceleration and Y movement acceleration

return

【図４５】

[Figure 45]



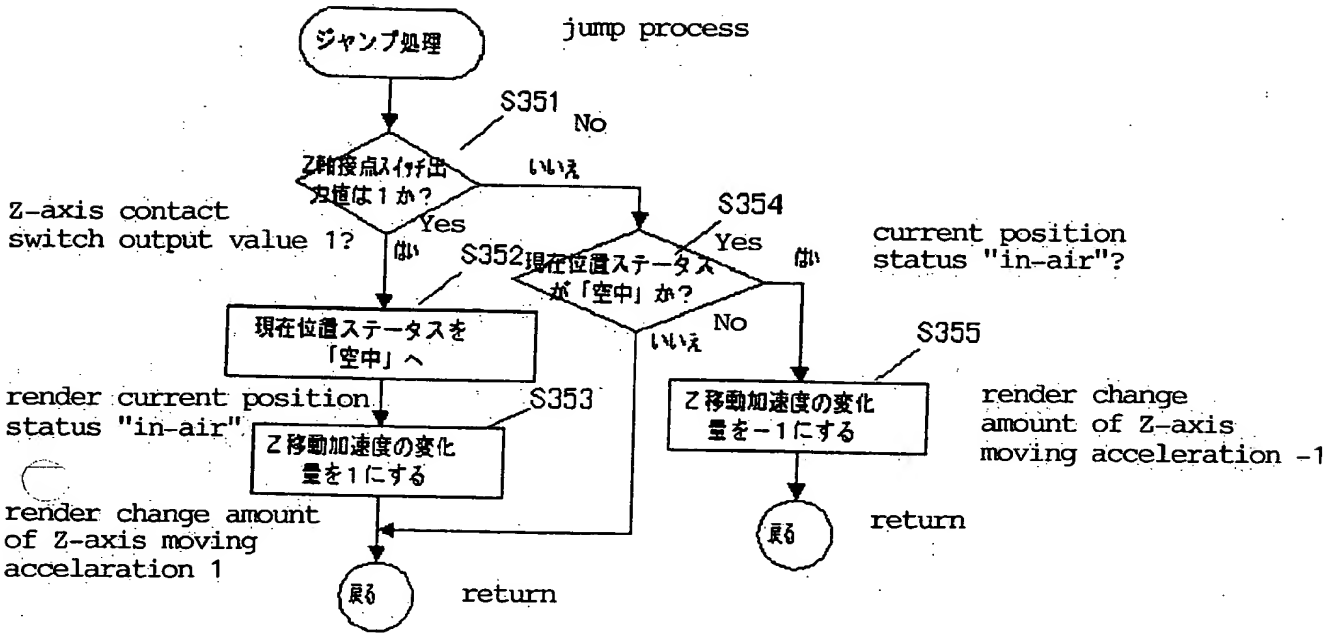
impact movement process

select conversion table

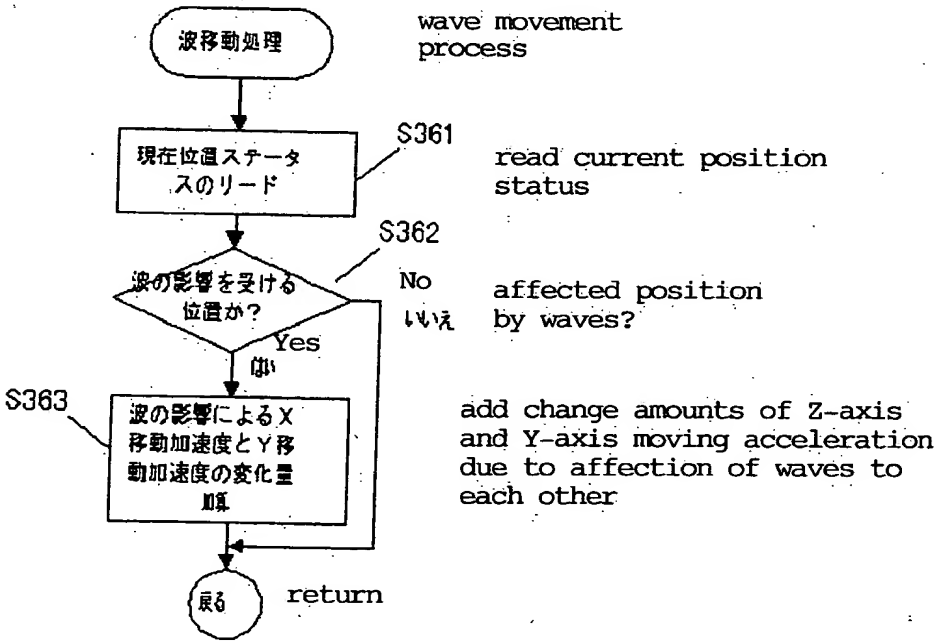
refer to conversion table and increase change amount in X movement acceleration and Y movement acceleration

return

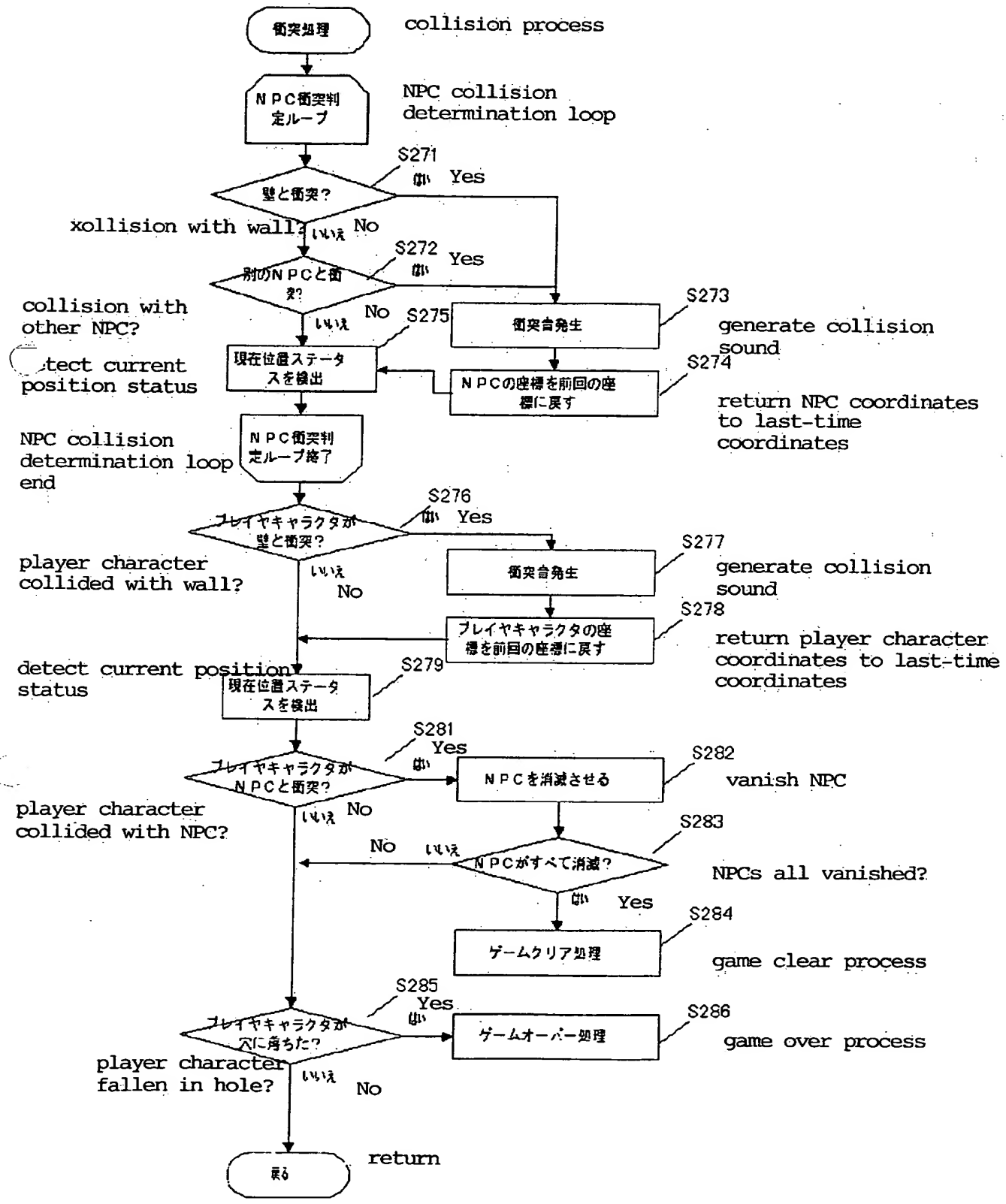
【図４６】 [Figure 46]



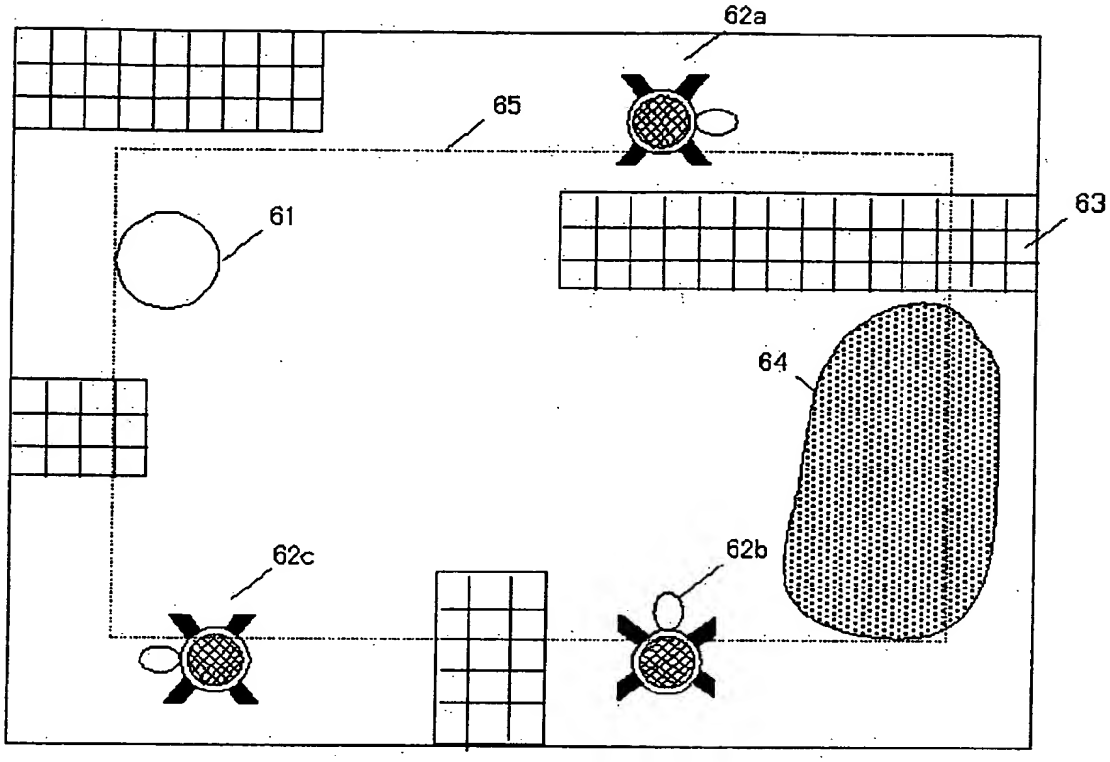
【図４７】 [Figure 47]



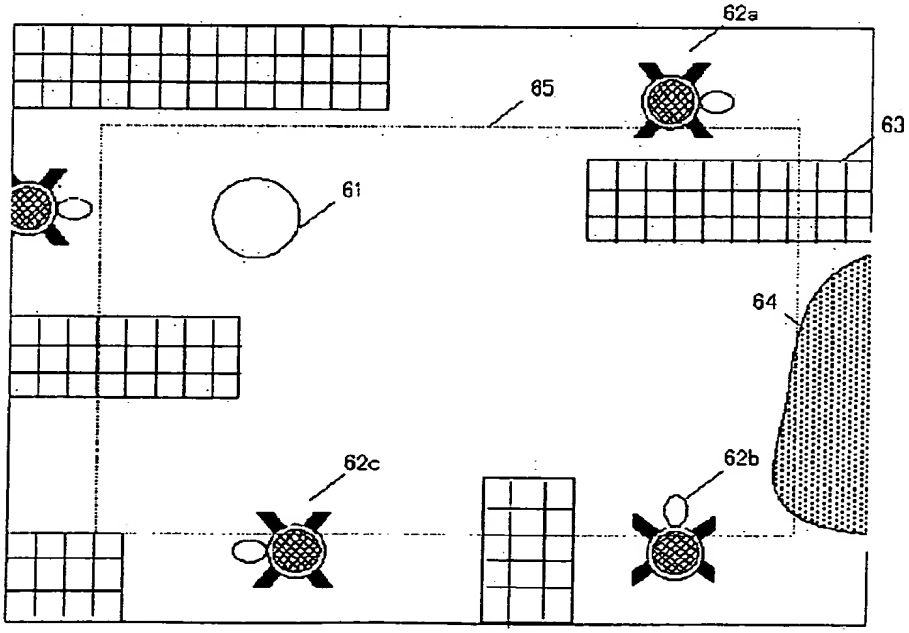
【図４８】 [Figure 48]



【図４９】 [Figure 49]



【図５０】 [Figure 50]



【図５１】 [Figure 51]

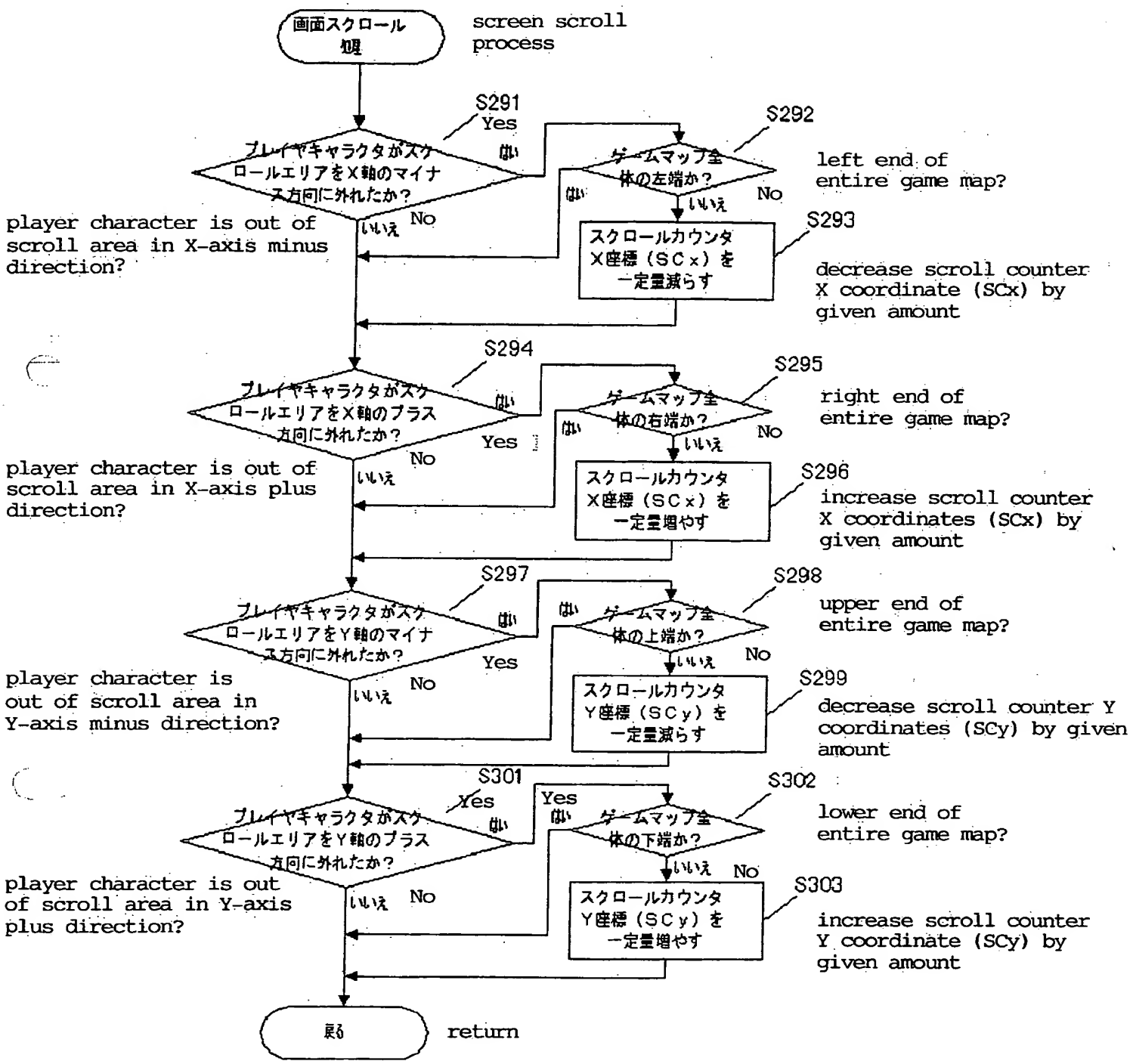


FIG. 20

game map select processing table

	utilization method	correction ratio	particular correction condition 1	particular correction number 1	particular correction condition 2	particular correction number 2
sensor output value X (INx)	change amount of X moving acceleration (Cx)	×2	–	–	–	–
sensor output value Y (INy)	change amount of Y moving acceleration (Cy)	×2	–	–	–	–
Z-axis contact SW output vlue (INz)	map decision	–	–	–	–	–
impact input flag (FS)	–	–	–	–	–	–

FIG. 21

player character moving table (in-air)

	utilization method	correction ratio	particular correction condition 1	particular correction number 1	particular correction condition 2	particular correction number 2
sensor output value X (INx)	–	–	–	–	–	–
sensor output value Y (INy)	–	–	–	–	–	–
Z-axis contact SW output vlue (INz)	change amount of Z moving acceleration (dAz)	×1	–	–	–	–
impact input flag (FS)	–	–	–	–	–	–

FIG. 22

player character moving table (on-floor)

	utilization method	correction ratio	particular correction condition 1	particular correction number 1	particular correction condition 2	particular correction number 2
sensor output value X (INx)	change amount of X moving acceleration (dAx)	×2	Inx>20	40	–	–
sensor output value Y (INy)	change amount of Y moving acceleration (dAy)	×2	Iny>20	40	–	–
Z-axis contact SW output vlue (INz)	change amount of Z moving acceleration (dAz)	×1	–	–	–	–
impact input flag (FS)	change amount of XY moving acceleration (dAx, dAy)	×3	–	–	–	–

FIG. 23

player character moving table (on-ice)

	utilization method	correction ratio	particular correction condition 1	particular correction number 1	particular correction condition 2	particular correction number 2
sensor output value X (INx)	change amount of X moving acceleration (dAx)	×3	Inx>20	60	–	–
sensor output value Y (INy)	change amount of Y moving acceleration (dAy)	×3	Iny>20	60	–	–
Z-axis contact SW output vlue (INz)	change amount of Z moving acceleration (dAz)	×1	–	–	–	–
impact input flag (FS)	change amount of XY moving acceleration (dAx, dAy)	×5	–	–	–	–

FIG. 24

player character moving table (under-water)

	utilization method	correction ratio	particular correction condition 1	particular correction number 1	particular correction condition 2	particular correction number 2
SENSOR OUTPUT VALUE X (INx)	change amount of X moving acceleration (dAx)	×1/2	Inx>20	60	-	-
SENSOR OUTPUT VALUE Y (INy)	change amount of Y moving acceleration (dAy)	×1/2	Iny>20	60	-	-
Z-AXIS CONTACT SW OUTPUT VLUE (INz)	change amount of Z moving acceleration (dAz)	×1	-	-	-	-
IMPACT INPUT FLAG (FS)	-	-	-	-	-	-

FIG. 25

NPC moving table (for tortoise normal position)

	utilization method	correction ratio	particular correction condition 1	particular correction number 1	particular correction condition 2	particular correction number 2
sensor output value X (INx)	change amount of X moving acceleration (dAx)	×1/2	Inx<10	0	Inx>20	10
sensor output value Y (INy)	change amount of Y moving acceleration (dAy)	×1/2	Iny<10	0	Iny>20	10
Z-axis contact SW output vlue (INz)	position inversion	-	-	-	-	-
impact input flag (FS)	-	-	-	-	-	-

FIG. 26

NPC moving table (for tortoise upside-down position)

	utilization method	correction ratio	particular correction condition 1	particular correction number 1	particular correction condition 2	particular correction number 2
sensor output value X (INx)	change amount of X moving acceleration (dAx)	×2	Inx>20	40	-	-
sensor output value Y (INy)	change amount of Y moving acceleration (dAy)	×1	Iny>20	40	-	-
Z-axis contact SW output vlue (INz)	position inversion	-	-	-	-	-
impact input flag (FS)	-	-	-	-	-	-